SERI/STR-217-3163
UC Category: 60
DE87001176

Lifting Surface Performance Analysis for Horizontal Axis Wind Turbines

D. Kocurek
Computational Methodology Associates
2900 Steve Drive
Hurst, Texas

SERI Technical Monitor:
J. Tangler

June 1987

Prepared under Subcontract No. IL-5-05105-1

Solar Energy Research Institute
A Division of Midwest Research Institute

1617 Cole Boulevard
Golden, Colorado 80401-3393

Prepared for the
U.S. Department of Energy
Contract No. DE-AC02-83CH10093

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America
Available from:
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Microfiche A01
Printed Copy A14

Codes are used for pricing all publications. The code is determined by the number of pages in the publication. Information pertaining to the pricing codes can be found in the current issue of the following publications, which are generally available in most libraries: *Energy Research Abstracts, (ERA); Government Reports Announcements* and Index (*GRA* and I); *Scientific and Technical Abstract Reports (STAR);* and publication, NTIS-PR-360 available from NTIS at the above address.

LIFTING SURFACE PERFORMANCE ANALYSIS FOR HORIZONTAL AXIS WIND TURBINES

SUMMARY

The application of numerical lifting surface theory to the calculation of a horizontal axis wind turbine's aerodynamic characteristics and performance is developed, and its computer implementation as Program LSWT (Lifting Surface Wind Turbine) is described. The method employed has evolved from rotary wing and helicopter applications, and features a detailed prescribed wake. The wake model is extended from a hovering rotor experimental generalization by including the effect of the windmill brake state on the radial and axial displacement rates of the trailing vortex system.

Performance calculations are made by coupling the lifting surface circulation solution to a blade element analysis that incorporates two-dimensional airfoil characteristics as functions of angle of attack and Reynolds number. Several analytical stall models are also provided to extend the airfoil characteristics beyond the limits of available data.

Although the primary emphasis of the current work is on the steady performance problem, the method includes facilities to investigate the effects of wind shear profile, tower shadow, and off-axis shaft alignment. These influences are included in a quasi-steady sense, and also add additional displacement detail to the wake model.

Correlation of the method to measured wind turbine performance and comparisons to blade element momentum theory calculations provide validation, but also highlight the extreme sensitivity of predictions to the quality of early post stall airfoil behavior.

TABLE OF CONTENTS

	Page
SUMMARY	ii
LIST OF FIGURES	iv
NOMENCLATURE	vi
INTRODUCTION	1
EXISTING METHODOLOGY	2
LIFTING SURFACE METHOD WITH PRESCRIBED WAKE	4
NUMERICAL LIFTING SURFACE THEORY	6
PANEL METHOD SOLUTION	6
CALCULATION OF AERODYNAMIC INFLUENCE COEFFICIENTS	12
COMBINED LIFTING SURFACE - BLADE ELEMENT THEORY	18
LIFTING SURFACE BLADE MODEL	20
BLADE SEGMENTATION	20
REFERENCE BLADE DEVELOPMENT	20
GENERALIZED PRESCRIBED WAKE FOR AXIAL FLOW	28
ROTOR WORK STATE EFFECT ON WAKE RATES	32
ROLLUP OF CONCENTRATED VORTICES	40
PERFORMANCE CALCULATION	44
SIMULATION OF POST-STALL AIRFOIL CHARACTERISTICS	49
VITERNA STALL MODEL	50
CMA STALL MODEL	51
EXTENSIONS FOR QUASI-STEADY SOLUTION	55
COMPARISON TO PERFORMANCE MEASUREMENTS	58
CONCLUSIONS AND RECOMMENDATIONS	71
REFERENCES	73
APPENDIX A, LSWT INPUT SEQUENCE GUIDE	A-1
APPENDIX B, PROGRAM AND INSTALLATION NOTES	B-1
APPENDIX C, PROGRAM LSWT SOURCE LISTING	C-1
APPENDIX D, SAMPLE INPUT AND OUTPUT FILE LISTINGS	D-1

LIST OF FIGURES

Figure		Page
1.	Horizontal Axis Wind Turbine Configuration and Reference Geometry Description	7
2.	Schematic of General Surface Panel and Wake Strip Indexing Scheme	11
3.	Coordinate System for Application of the Biot-Savart Law to a Linear Vortex Element	13
4.	Coordinate System for Modeling of the Far Wake With a Semi-Infinite Vortex Cylinder	16
5.	Development of Reference Blade Geometry From Segment Descriptions of Major Components	21
6.	Spanwise Panel Distributions Corresponding to Blade Segment Grid Options	22
7.	Modeling of Complex Planform With Segment Chord Reference Station $(x/c)_{ref}$ and Distributed Planform Offset	25
8.	Association of Airfoil Characteristic Tables with Segment Spanwise Regions Allows Interpolation Between Tables	26
9.	Gray's Concept of Hovering Rotor Wake Structure	29
10.	Model Rotor Tip Vortex Trajectory Measured From Schlieren Flow Visualization Photographs of Wake (Reference 3)	31
11.	Rotor Work State Velocity Relationships in Axial Flow	36
12.	Rotor Work State Settling Rate Relationships in Axial Flow	39
13.	Schematic of Wake Grid Regions, Sheet Definitions, and Rollup Options	42
14.	Panel, Panel Boundary Grid, and Wake Trailer Index Assignments	43
15.	Flow Chart of Major Steps in Solution Procedure	45
16.	Velocity, Force Coefficient, and Angle of Attack Resolution at a Blade Element	47

17. CMA Stall Model Region Definitions and Key Points for Matching Data 52

18. Comparison of Calculated and Measured Rotor Power, NASA MOD-0 Wind Turbine, Viterna Stall Model 61

19. Comparison of Calculated and Measured Rotor Power, ESI-54 Wind Turbine, Viterna Stall Model 62

20. Comparison of Calculated and Measured Rotor Power, Carter 25 Wind Turbine, Viterna Stall Model 63

21. Comparison of Calculated and Measured Rotor Power, Enertech 1500 Wind Turbine, Viterna Stall Model 64

22. Comparison of Calculated and Measured Rotor Power, NASA MOD-0 Wind Turbine, CMA Stall Model 65

23. Comparison of Calculated and Measured Rotor Power, ESI-54 Wind Turbine, CMA Stall Model 66

24. Comparison of Calculated and Measured Rotor Power, Carter 25 Wind Turbine, CMA Stall Model 67

25. Comparison of Calculated and Measured Rotor Power, Enertech 1500 Wind Turbine, CMA Stall Model 68

NOMEMCLATURE

Symbol	Definition	Dimension
A	minimum nondimensional wake radius	
AR	aspect ratio	
A_1, A_2, B_1, B_2	coefficients of Viterna stall model equations	
A_{12}, A_{22}, B_{12}, B_{22}, A_{13}, A_{23}, B_{13}, B_{23}	coefficients of CMA stall model equations	
B	general boundary surface	
B_{pr}	base pressure factor (post stall)	
$\overline{C}$, C	contour direction unit vector, contour path	
C_T	rotor thrust coefficient	
$\overline{D}$, D	vector, scaler aerodynamic influence coefficients	ft^{-1}
D^*	aerodynamic influence coefficient, modified for non-potential lift response	ft^{-1}
$\tilde{D}$	aerodynamic influence coefficient, modified for quasi-steady effects	ft^{-1}
K	distributed circulation	ft^2/sec
$\tilde{K}$	ratio of disk average to target blade distributed circulation	
$\overline{N}$	positive normal vector	
P	discretized surface panel	
R	rotor radius; also used to nondimensionalize all geometry parameters	ft
S	lifting surface	
T	rotor thrust	lb

T_s	tower shadow velocity deficit fraction	
T1, T2, T3, T4	intermediate terms in vortex cylinder calculations	
V_a	axial impressed velocity	ft/sec
$\overline{V}_a$	referred axial impressed velocity	
V_{net}	net axial velocity through rotor	ft/sec
$\overline{V}_{net}$	referred net axial velocity through rotor	
V_r	blade element relative velocity	ft/sec
V_w	wind profile speed	ft/sec
V_{w_0}	reference wind speed	ft/sec
$\overline{V}_\infty$	freestream velocity vector	ft/sec
W	wake surface	
a	lift curve slope	rad^{-1}
a_e	effective linear lift curve slope	rad^{-1}
c	local blade chord	ft
c_d	airfoil drag coefficient	
$c_{d_{max}}$	airfoil maximum drag coefficient (post stall)	
c_{d_s}	airfoil drag coefficient at stall	
$c_{d_{45}}$	airfoil drag coefficient at 45 degrees	
$c_{d_{90}}$	airfoil drag coefficient at 90 degrees	
c_l	airfoil lift coefficient	

c_{l_r}	airfoil lift coefficient at recovery angle	
c_{l_s}	airfoil lift coefficient at stall	
$c_{l_{45}}$	airfoil lift coefficient at 45 degrees	
$c_{l_{90}}$	airfoil lift coefficient at 90 degrees	
c_m	airfoil moment coefficient	
c_s	leading edge suction force coefficient	
c_x, c_y, c_z	blade element aerodynamic force coefficient components	
c_x, c_y, c_z	induced velocity unit direction vector components	
e	wind profile exponent	
g_0	leading edge suction force parameter	
h	hub (shaft) height above ground	ft
i,j,k	Cartesian unit direction vector components	
k_a	wake axial settling rate due to impressed velocity transport	
$\bar{k}_a$	referred wake axial settling rate due to impressed velocity transport	
k_i	wake axial settling rate due to mean axial induced velocity transport	
k_r	radial contraction rate	
k_{r_h}	hover reference radial contraction rate	
k_z	wake axial settling rate	
$\bar{k}_z$	referred wake axial settling rate	
k_{z_h}	hover reference axial settling rate	

n_s	number of azimuthal integration steps in vortex cylinder calculations	
$\overline{q}$	vector induced velocity	ft/sec
$\overline{r}$, r	radial vector, radial coordinate	ft
v	mean axial induced velocity	ft/sec
$\overline{v}$	referred mean axial induced velocity	
v_x, v_z	blade element induced velocity components	ft/sec
v_h	mean hover axial induced velocity	ft/sec
x,y,z	Cartesian coordinates	ft
$(x/c)_{ref}$	chord reference station	
Γ_{te}	bound circulation	ft^2/sec
Λ	aerodynamic sweep angle	rad
Ω	rotor rotational speed	rad/sec
α	general angle of attack	rad
α_e	effective angle of attack	rad
α_g	geometric angle of attack	rad
α_i	induced angle of attack	rad
α_n	inflow angle of attack	rad
α_r	recovery angle of attack (post stall)	rad
α_{s_d}	drag stall angle of attack	rad
α_{s_l}	lift stall angle of attack	rad
α_w	wind angle of attack	rad
α_0	zero lift angle of attack	rad

α_{s_1}	horizontal shaft offset with respect to reference wind direction	rad
α_{s_2}	vertical shaft offset with respect to reference wind direction	rad
λ	wake radial contraction parameter	
λ_h	hover reference radial contraction parameter	
ψ	azimuth coordinate	rad
ψ_w	wake azimuth coordinate	rad
ψ_{mr}	azimuth increment over which inboard sheet forms into root vortex	rad
ψ_{mt}	azimuth increment over which outboard sheet forms into tip vortex	rad
ϕ	azimuthal integration variable in vortex cylinder calculations	rad
ρ	atmospheric density	$slug/ft^3$

Indices, Superscripts, and Subscripts

l	target point
k	source point
N	normal direction to surface
i	lifting surface panel row index
j	lifting surface panel column index, wake strip index
le	leading edge
m	maximum row index
n	maximum column index, or blade index
s	lifting surface
te	trailing edge
w	wake
1,2,3	local blade segment coordinate systems

a,b,c	field points
x,y,z	Cartesian vector components
r	radial component

INTRODUCTION

Wind energy conversion systems based on horizontal axis rotor configurations have evolved through the stages of technology demonstration to the level of a mature concept which now demands more rigorous methodology for analysis and design of an efficient, cost effective machine. In particular, a reliable, working methodology for aerodynamic design and performance prediction is a key technology element required to guide further development in this area.

Recent progress in the area of helicopter rotor hover performance analysis as described in References 1 through 3, has demonstrated a high level of predictive capability using numerical lifting surface theory combined with a distorted prescribed wake model. The objective of the present work is to extend applicable elements of this similar technology base to wind energy turbines and establish an aerodynamics methodology from which configuration concepts for performance benefits can be achieved.

This report describes the lifting surface method and the development of a generalized prescribed wake for wind turbine application. The method includes a comprehensive yet straightforward procedure to model complex blade geometry, a wake model which allows roll-up of tip and root vortices, and facilities for investigating quasi-steady effects of wind shear profile, tower shadow, and off-axis operation.

Detailed blade aerodynamics and system performance are calculated by combining the lifting surface method with a blade element analysis that incorporates two-dimensional airfoil lift, drag, and moment coefficients as functions of angle of attack and Reynolds number. High angle of attack analytical simulations are used to extend the supplied airfoil characteristics into stall. The method also incorporates a procedure to guarantee compatibility between the lifting surface circulation and the nonlinear sectional response impressed through the airfoils. Also, the use of lead-

ing edge suction forces in the calculation of the radial distribution of induced drag affords the most accurate prediction of torque, especially for low aspect ratio blades and those with advanced planforms or tips.

Implementation of the analysis as Fortran Program LSWT (Lifting Surface Wind Turbine) is also described, and an input guide, programming notes, source code listing, and examples of input and output files are provided as appendices to this report.

Examples of correlation of the present method to measured performance for one research and three commercial wind turbines demonstrate and validate the analysis. For all cases presented, calculations are also shown from a blade element momentum analysis. In constructing the cases for comparing the two analytical methods, the same blade geometry description, placement of solution stations, airfoil characteristics, and stall models were used down to the code level where possible. The lifting surface results show generally good agreement with measurements of power output at low to moderate wind speeds, and are consistently higher than the momentum analysis. At the wind speed for peak power and beyond the difference grows significantly, the lifting surface results still being higher. In this region the extreme sensitivity of the results to modeling of early post stall airfoil behavior is shown; however, with reasonable assumptions for the airfoil post stall characteristics the lifting surface method provides improved predictions of the peak power levels.

EXISTING METHODOLOGY

Because of experienced and knowledgeable users, analytical tools have always been available for evaluating limited modifications to established configurations. Such methods often contain empirical corrections designed to produce satisfactory performance correlation over an available data base. This is certainly the case with the traditional axial flow blade element momentum theory techniques which find wide use throughout rotary-wing analysis, and which are in popular use for wind system applications. Properly used, momentum theory is very valuable for rapid estimation of

integrated performance with minimal impact on computational resources. However, inherent assumptions in the theory limit its usefulness in producing detailed blade aerodynamic distributions for diagnostic or design information.

Originally derived for analysis of cruising propellers with very slight induced velocity, these procedures are more frequently applied to rotors in hover, and low speed climb or descent. In most applications adjustments to the theory are necessary, especially to provide correct levels of induced power. These corrections may be categorized as either modifying the radial distribution of induced velocity or the overall magnitude of the induced velocity.

Accuracy of Radial Loading

The category of corrections dealing with the modification of the radial induced velocity is the most familiar and includes the array of methods to approximate the physical boundary condition of zero loading at the streamwise edges, or tips of lifting blades. Most procedures are derived with various degrees of simplification from the original work of Prandtl and Betz as best described by Glauert (Reference 4). In concept the correction accounts for "tip losses" associated with a finite number of blades. That is, momentum theory always assumes an infinite number of lightly loaded blades of infinite aspect ratio, and thus non-zero loading at blade tips.

This assumption also equates to azimuthwise uniform induced velocity. Actually, there is an impulse of induced velocity from each blade's trailed wake causing the velocity at the blades to be higher than the disk average. It is also a particular blade's wake that causes that blade's loading to smoothly approach zero at the tip. However the tip loss approximation does not completely account for this effect since the tip loading is shaped by the blades finite planform as well. In this sense the tip loss approximates finite span, but cannot account for finite blade aspect ratio. Lifting line theory accounts for finite span, but still assumes infinite

aspect ratio. Blade planform effects can only be modeled with lifting surface theory, and accordingly, the emphasis of the analysis shifts from one of only predicting integrated performance to that of additionally calculating detailed distributions of aerodynamic parameters.

Influence of the Rotor's Vortex Wake

The second type of correction is needed to account for deviations in the true momentum flux from the ideal model. In actual flows the theoretical slipstream acceleration of axial induced velocity is never realized. This is because of radial leakage across the real wake boundary, the spiraling tip vortex trajectory, and diffusion mechanisms which cause dissipation of the wake cylinder. Thus, for a given level of thrust the axial impulse of induced velocity produced at the rotor disk must be higher than the theoretical in order to establish the momentum flux from far upstream to far downstream.

A correction for this effect is included in the empirical rotor work state models used to describe the mean velocity through the rotor as a function of the axial freestream velocity. These models have been derived, however, by setting the apparent flow through the rotor in either a disk or blade element momentum analysis to give agreement with measurements of thrust and power. The result is that a particular model is then strictly consistent only with the analysis used in its derivation, with assumed airfoil characteristics and profile drag levels, with the tip loss model, and with the accuracy of the measurements. References 5 and 6, describe wind turbine work state models based on Glauert's work. Although the importance of this consideration reduces with wind speed, Castles and Gray show striking differences and comment that Glauert's empirical formula is highly suspect at lower wind speeds (rotor descent rates) as the vortex ring state is approached.

LIFTING SURFACE METHOD WITH PRESCRIBED WAKE

Lifting surface theory with a realistic wake model thus reduces the

uncertainties discussed by eliminating the need for the tip loss correction, and by providing an inflow model with physical details. Further, the access that such an analysis provides for independently examining details leads to correct identification of modeling deficiencies and subsequent improvement.

In helicopter rotor aerodynamic analyses, the prescribed wake method has become the standard for performance prediction and blade design. Originally conceived by Gray (Reference 8), the approach was generalized by Landgrebe (Reference 9), and established by the broad experimental data base measured by Kocurek and Tangler (References 3). In the prescribed wake approach the complex distorted wake geometry is economically computed in terms of basic parameters which are themselves coupled to some primary variable of the analysis, such as thrust level. These coupling relationships are derived from experimental measurements of wake geometry.

Reference 10 describes the UTRC WECSPER analysis and shows the feasibility of the wake modeling approach for wind turbines. This method is extended from hovering rotor lifting line theory combined with a classical wake with modifications for nonuniform onset wind conditions. Although accommodations for a distorted wake are made in the analysis, no extensive evaluation of the required level of wake detail has been made. This is due in part to the lack of a systematically obtained experimental data base of wind turbine wake geometry.

The results obtained with the classical wake are encouraging and probably consistent with the relative pitch of the wake in hover compared to the wake additionally displaced by axial onset velocity. Also, the wind turbine's expanding wake places the tip vortices outside the rotor disk's perimeter, thus lessening their influence in comparison to their position for a rotor in hover. These observations suggest that wind turbine wake behavior is dominated by kinematic displacement and that distortions due to induced effects can be derived and superposed with sufficient accuracy from established techniques for prescribing hovering rotor wakes. This is then the approach taken in the present work.

NUMERICAL LIFTING SURFACE THEORY

The basic lifting surface method has evolved with emphasis on extended features which accommodate the special considerations for application to rotary wings such as summarized for calculation of helicopter hover performance in Reference 1. As illustrated in Figure 1, the horizontal axis rotor is described in a global Cartesian coordinate system with origin at the rotor hub, and aligned with the shaft. The shaft may have horizontal and vertical angular offsets from the principal wind direction. Positive rotation is in the right-handed sense, and a wind vector in the positive z-axis direction places the rotor in descent (normal turbine operation). Nose-up, positive, blade pitch and twist increment are defined when the blade leading edge is displaced in the positive z- direction. This system and convention are consistent with general rotor theory, and are required for self-consistent handling of the detailed geometry of the rotor lifting surface and wake model. This convention is opposite to most turbine analyses.

PANEL METHOD SOLUTION

The lifting surface theory is applied through a panel method formulation which is nonlinear in that all stream and induced velocity components are considered as well as a full three-dimensional thin surface representation of each blade. By using an influence coefficient formulation derived with vector analysis, a linear system is maintained without the usual neglect of radial flow at a blade or the assumption of small angles.

In the rotor fixed coordinate system, Laplace's equation governs the steady, irrotational flow, and an application of Greens's theorem (Reference 2) shows that all surfaces of discontinuity in the flow field may be replaced by a doublet sheet. For the present application all blades and trailing wakes are thus replaced. Further, the disturbance, or induced, velocity at point 1 due to the doublet sheet of net circulation strength K is

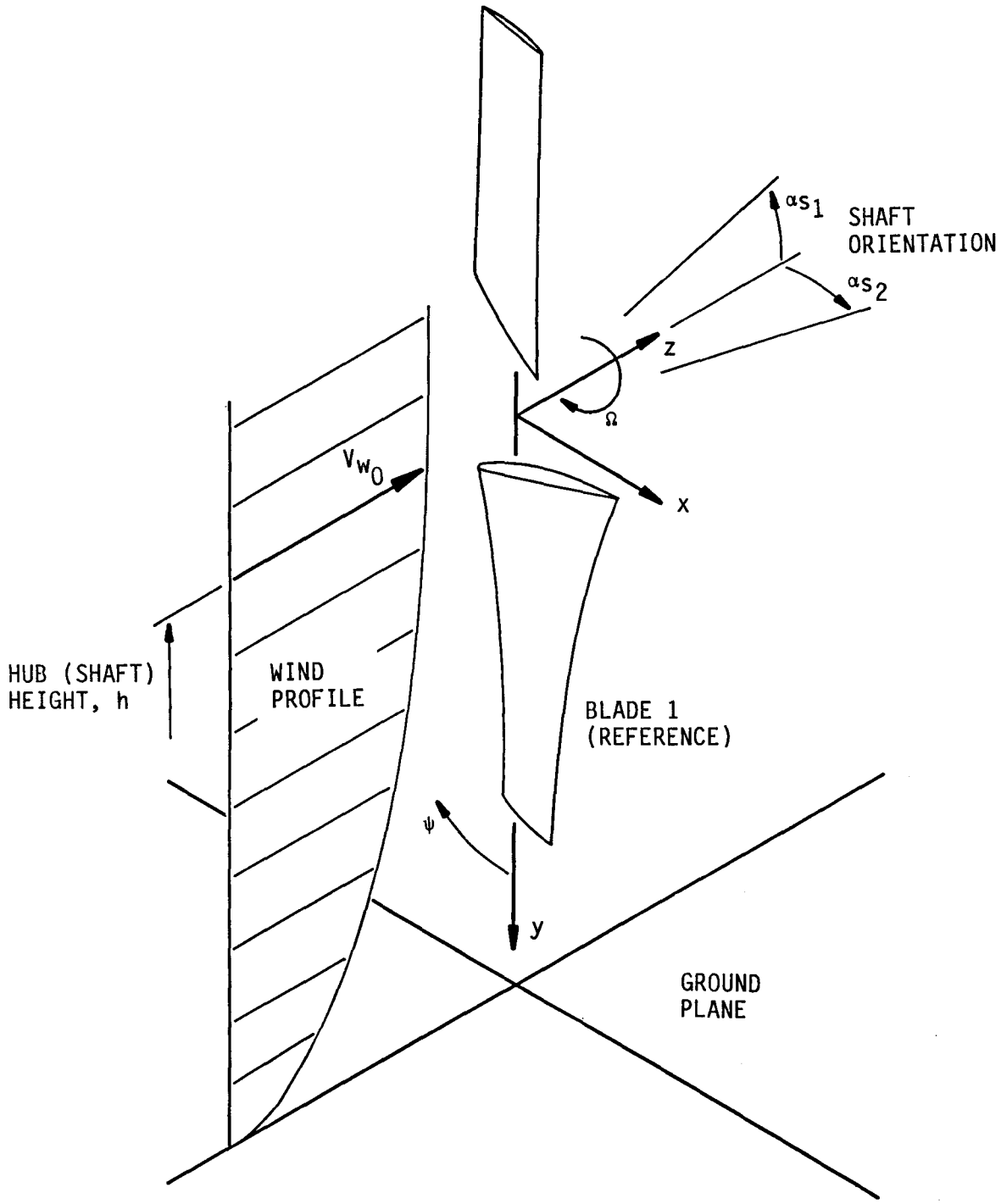


Figure 1. Horizontal Axis Wind Turbine Configuration and Reference Geometry Description.

$$\overline{q}_1 = \int_{S+W} \nabla_1 K \frac{\partial}{\partial \overline{N}} \frac{1}{4\pi r} dB \tag{1}$$

where the subscript of the gradient operator denotes differentiation with respect to the coordinate variables at 1, and $\overline{N}$ is the positive normal vector on the surface.

The blade and wake surface (S+W) is then subdivided into small regions, or panels, P_k, and the circulation is approximated as uniform over each. Then the surface integration may be taken discretely over each panel and the contributions summed,

$$\overline{q}_1 = \sum_k K_k \int_{P_k} \nabla_1 \frac{\partial}{\partial \overline{N}_k} \frac{1}{4\pi r} dB \tag{2}$$

The solution procedure can now be focused on calculating the vector aerodynamic influence coefficients (AIC's) defined by

$$\overline{D}_{1,k} = \int_{P_k} \nabla_1 \frac{\partial}{\partial \overline{N}_k} \frac{1}{4\pi r} dB \tag{3}$$

which give the induced velocity influence at (x_1, y_1, z_1) per unit strength of the doublet panel P_k. Through Stoke's theorem the surface integration giving the AIC's is reduced to a contour integration about the perimeter C_k of P_k,

$$\overline{D}_{1,k} = \int_{C_k} - \frac{\overline{C} \times \overline{r}}{r^3} dC \tag{4}$$

where $\overline{C}$ is the unit vector along the contour. Equation (4) is simply a statement of the well known Biot-Savart law for a closed vortex filament of unit strength.

By building the panel geometry solely from straight line elements then the contour integration of Equation (4) can be conveniently broken into separate intervals over each circuit element, and the resulting quantities summed to give the total contribution of the panel. The integral taken over a straight line vortex filament has a simple analytic solution given entirely by the Cartesian coordinates of the target point 1 and the end point of the filament. Calculation of the AIC's will be described in a later section.

The unknowns of the problem are now the circulation strengths of each panel, and their solution is obtained by forming a set of simultaneous linear algebraic equations based on Equation (2) while satisfying flow tangency on the blade surfaces and the Kutta condition at the trailing edges. The formulation is reduced by noting that at a point within panel P_1 on a blade surface, flow tangency requires

$$\overline{N}_1 \cdot (\overline{V}_\infty + \overline{q}_1) = 0 \tag{5}$$

Then the normalwash velocity at the panel

$$q_{N_1} = \overline{N}_1 \cdot \overline{q}_1 = -\overline{N}_1 \cdot \overline{V}_\infty \tag{6}$$

and the scalar influence coefficient

$$D_{1,k} = \overline{N}_1 \cdot \overline{D}_{1,k} \tag{7}$$

in combination with Equations (2) yield the basic relation of the present theory

$$q_{N_l} = \sum_k D_{l,k} K_k \tag{8}$$

Since the wake must be force free, the circulation along a wake streamline is constant, and its value must be that of the circulation at the blade trailing edge at the point from which the wake streamline emanates. Enforcement of the Kutta condition at the trailing edge, which renders the blade surface circulation unique, further requires that the circulation be continuous. The wake panel idealization thus becomes a series of semi-infinite strips trailing aftward from the blade in the streamwise direction.

If an m by n array of panels are arranged on a blade surface as shown in Figure 2, in streamwise columns and spanwise rows, then associated with each surface column of panels is a wake strip. The indices i and j are used to identify the panel of the ith row and jth column, and the jth wake strip. The panels are numbered by rows so that panel P_l is indexed by

$$l = (i-1)n + j, \; 1 \leq l \leq (m \cdot n) \tag{9}$$

Equation (8) can be written for the l_{th} panel by taking separate summations for surface and wake contributions to the normalwash,

$$q_{N_l} = \sum_{k=1}^{(m \cdot n)} D_{s_{l,k}} K_k + \sum_{j=1}^{n} D_{w_{l,j}} K_{w_j} \tag{10}$$

Specific enforcement of the Kutta condition allows the wake strip strengths K_w to be expressed as functions of the surface strengths K_k thus reducing the unknowns to just the surface distribution. This is accomplished by assuming that $(K_w/K_m)_j$ is the same as the ratio known from the two-dimensional solution. Equation (10) is then simplified to

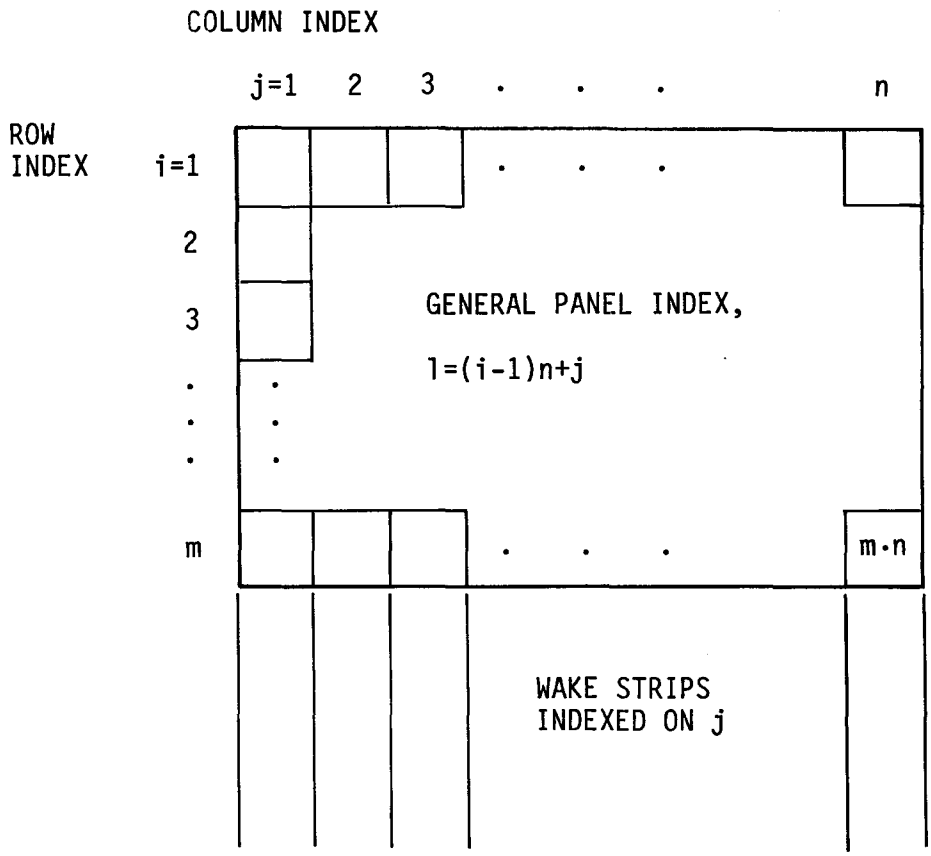


Figure 2. Schematic of General Surface Panel and Wake Strip Indexing Scheme.

$$q_{N_l} = \sum_{k=1}^{(m \cdot n)} (D_{sw})_{l,k} K_k \tag{11}$$

Applying this equation to a collocation point at each surface panel P_l on a target blade and accounting for the velocity induced by all source blades and wakes generates a system of (m•n) linear algebraic equations which are solved for the (m•n) unknown K_k. Thus, for given surface and wake geometry, the circulation produced in response to surface motion can be obtained.

CALCULATION OF AERODYNAMIC INFLUENCE COEFFICIENTS

The working equations for calculation of the AIC's are presented in this section. The primary calculations are for the discrete straight line vortex geometry of the idealized blade and wake surfaces. The far wake is optionally approximated by a vortex cylinder. In the following it is understood that coordinate variables have been normalized by unit radius R.

Line Vortex Induced Velocity

The following implementation of the Biot-Savart law in Cartesian coordinates is used to calculate the contributions to the AIC's from each vortex element. This formulation is derived from parametric integration of the governing vector equation. A source element is considered to run from point 'a' to point 'b' such that the direction from a to b is consistent with the right hand rule in view of the element's circulation sense. Point 'c' is the target point where the velocity is calculated. This geometry is sketched in Figure 3. With the parameter definitions

$$A = (x_a-x_c)^2 + (y_a-y_c)^2 + (z_a-z_c)^2$$
$$B = 2[(x_a-x_b)(x_c-x_a) + (y_a-y_b)(y_c-y_a) + (z_a-z_b)(z_c-z_a)]$$
$$C = (x_a-x_b)^2 + (y_a-y_b)^2 + (z_a-z_b)^2$$

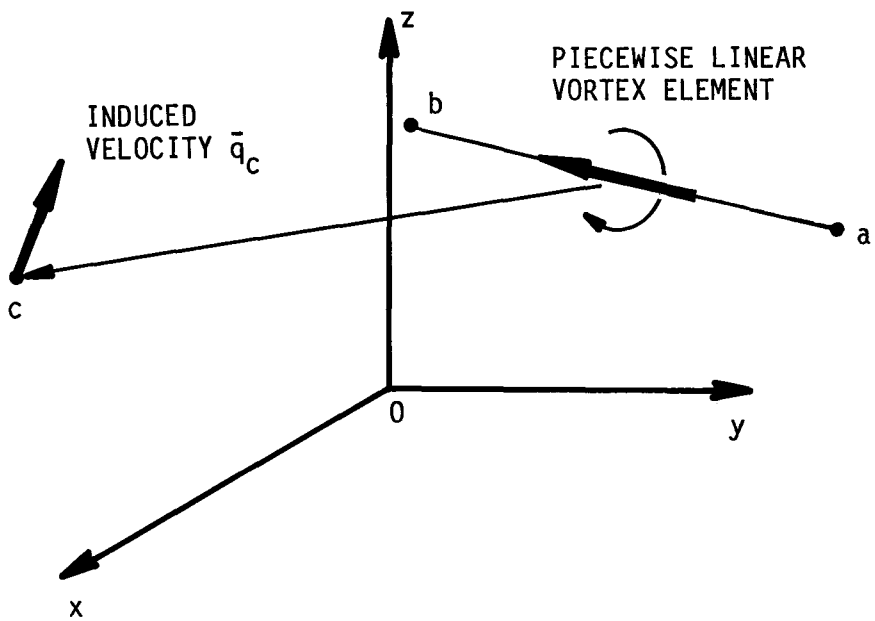


Figure 3. Coordinate System for Application of the Biot-Savart Law to a Linear Vortex Element.

and

$$Q = 4AC - B^2 \tag{12}$$

then the magnitude of the velocity induced at point c per unit vortex strength is for $Q \neq 0$,

$$|\overline{q}| = [1/(2\pi Q)][(2C+B)/(A+B+C)^{1/2} - B/A^{1/2}] \tag{13}$$

If $Q = 0$ then point c is collinear with a and b, and the velocity induced is zero. The cartesian vector components of the induced velocity are obtained by multiplying the velocity magnitude by the appropriate direction vector components which from the defined element geometry are

$$\begin{aligned} c_x &= (y_c-y_b)z_a + (y_a-y_c)z_b + (y_b-y_a)z_c \\ c_y &= (z_c-z_b)x_a + (z_a-z_c)x_b + (z_b-z_a)x_c \\ c_z &= (x_c-x_b)y_a + (x_a-x_c)y_b + (x_b-x_a)y_c \end{aligned} \tag{14}$$

giving

$$\overline{q} = |\overline{q}|[\ (c_x)i + (c_y)j + (c_z)k\ ] \tag{15}$$

Having now obtained the vector velocity induced by the element at the target point, the element's contribution to its surface panel or wake strip AIC is simply the scalar product of the velocity and the direction vector for the desired influence coefficient sense. This formulation is easily applied to the discrete geometry of the surface and wake model by systematically sweeping the coordinate arrays, calculating the influence of each straight line element in turn, while summing each element's contribution to the applicable AIC's.

Vortex Cylinder Induced Velocity

An optional analytical far wake is included in the analysis and modeled by extending each filament of the specified length discretized wake with a continuous, semi-infinite vortex cylinder. The influence of a vortex cylinder is calculated following the derivation presented by Miller in Reference 11, except that direct numerical integration is used exclusively instead of in combination with elliptic integrals. Direct integration is chosen since the number of steps required for accuracy is similar to the number of terms needed in the usual series representation of the elliptic integrals, and implementation is more straightforward.

Figure 4, shows the relative geometry of the target point c and the semi-infinite vortex cylinder of radius r_a. The mouth of the cylinder is located at the axial coordinate z_a. Following Miller's derivation, the axial and radial induced velocities due to a cylinder of unit strength density are given by

$$q_z = \frac{1}{4\pi R}\int_0^{2\pi} \frac{r_a(r_a - r_c \cos\phi)}{r_c^2 + r_a^2 - 2r_a r_c \cos\phi} \cdot \left(1 - \frac{z_a}{(r_c^2 + r_a^2 + z_a^2 - 2r_a r_c \cos\phi)^{1/2}}\right) d\phi \qquad (16)$$

$$q_r = \frac{1}{4\pi R}\int_0^{2\pi} \frac{r_a \cos\phi}{(r_c^2 + r_a^2 + z_a^2 - 2r_a r_c \cos\phi)^{1/2}}\, d\phi \qquad (17)$$

The tangential, or swirl, component due to the far wake is assumed to be negligible.

A simple numerical integration procedure for these expressions divides the interval into a number of steps which vary with the axial distance from

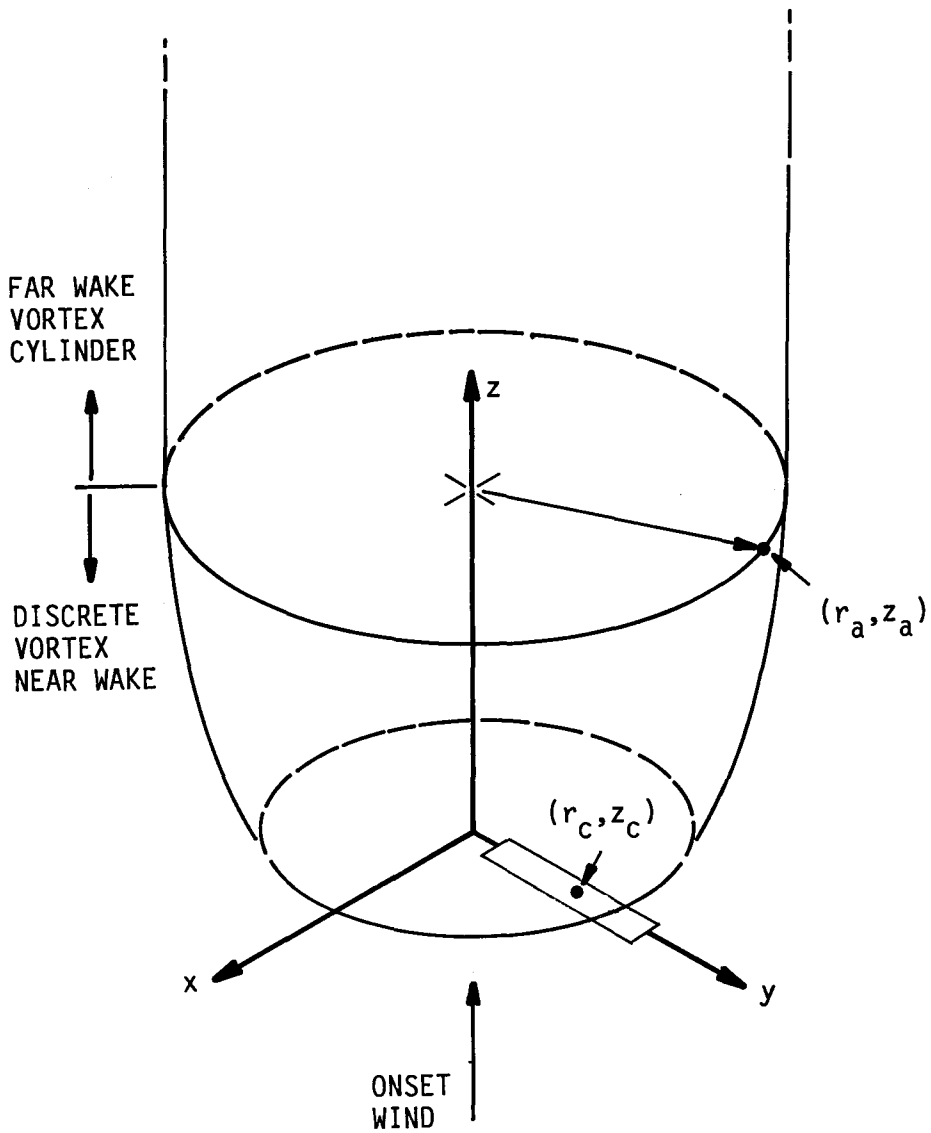


Figure 4. Coordinate System for Modeling of the Far Wake With a Semi-Infinite Vortex Cylinder.

the target point to the mouth of the vortex cylinder. If the distance is greater than unit radius then ten steps are used. Otherwise, the number of steps is calculated from

$$n_s = 100 - 90(|z_a - z_c|) \tag{18}$$

The step size is then $\Delta\phi = 2\pi/n_s$. If the value of ϕ is taken as the average over a step, and with the following definitions

$$T1 = r_a - r_c \cos\phi$$

$$T2 = r_c^2 + r_a^2 - 2r_a r_c \cos\phi$$

$$T3 = (r_a^2 + r_c^2 + (z_a - z_c)^2 - 2r_a r_c \cos\phi$$

$$T4 = \frac{\Delta\phi}{8\pi^2 k_z} \tag{19}$$

the cylindrical components of the induced velocity are then simply summed as

$$q_z = T4 \sum_i r_a (T1/T2)[1-(z_a - z_c)/T3] \tag{20}$$

$$q_r = T4 \sum_i r_a \cos\phi/T3 \tag{21}$$

The term T4 contains the axial settling rate k_z of the near wake, and thus sets the strength density of the vortex cylinder based on the pitch of the vortex spiral which feeds it.

The cartesian components of the radial induced velocity are resolved as

$$q_x = q_r(x_c/r_c) \tag{22}$$

$$q_y = q_r(x_c/r_c) \tag{23}$$

which together with the previously defined z-component yield the required AIC when combined as a scalar product at the target point c with its direction vector.

COMBINED LIFTING SURFACE - BLADE ELEMENT THEORY

A feature unique to the present method is a technique termed circulation response tailoring. This modification replaces the inherent linear sectional lift response characteristics of the potential flow theory with those derived from supplied two-dimensional airfoil coefficients. These coefficients are also used in a blade element strip integration to obtain performance characteristics. Thus, the loading from the circulation solution is fully compatible with the blade element performance analysis.

Inherent in any potential flow analysis is that the lift, or equivalently circulation, response to angle of attack is through the lift slope 2π. Consider a section of the lifting surface to be responding to an effective angle of attack defined relative to the local freestream. The effective angle of attack is the net result of the combined interference from all other sections of the lifting surface and wake, and defines the normalwash contribution for satisfying flow tangency which must be supplied by the self-induced velocity of a section. The section circulation which develops is that which will satisfy this partial velocity boundary condition. The circulation can be tailored then by modifying the self-induced velocity to preserve the boundary condition while allowing prescribed rates of circulation to develop.

The full derivation of this procedure is in Reference 2. Aerodynamic influence coefficients are calculated which give the self-induced velocity. These are actually just two-dimensional AIC's for the simple flat plate case. With this AIC denoted as D_{1s}, the self-induced velocity at a panel can be isolated, modified, and then recombined to form the final coefficient which will produce a sectional response with a lift slope of a_j. The modified response coefficient is

$$D^*_{l,k} = D_{l,k} + \left[\frac{2\pi}{a_j} - 1\right](D_{ls})_k \tag{24}$$

when a source panel is in the same column as the target,

$$j(k) = j(l) \tag{25}$$

and otherwise remains unchanged. In application the sectional lift slopes are nonconstant functions of angle of attack, and thus an iteration is required to produce consistency between the calculated circulation, which determines effective angle of attack, and the corresponding effective linear lift slope.

LIFTING SURFACE BLADE MODEL

The modeling and specification of blade geometric characteristics for development of the lifting surface panel grid is designed to provide maximum flexibility for a wide variety of configurations while maintaining simplicity of input data requirements. The goal is to have any restrictions be due to technical assumptions in the physical model rather than to any inability to represent the blade geometrically.

BLADE SEGMENTATION

To accomplish this, the concept of a segmented blade is adopted. By describing the blade as a series of spanwise segments, abrupt changes or even discontinuities in geometry or airfoil definition can be faithfully reproduced in the lifting surface idealization. Figure 5 shows a blade modeled with cuff, main blade body, and tip segments. As many as four completely independent segments can be utilized. This limit is selected only to set program array dimensions, and can of course be easily changed. Each segment is also described independently in the input data set. Thus, the input for the simplest case of a single segment blade is not complicated by this capability. For more than one segment, the geometric description is repeated for each in a completely parallel format.

Surface Panel Arrangement

The number and spanwise arrangement of panels on the blade surface is also specified individually for each blade segment, and is controlled through four grid options as illustrated in Figure 6. Within each segment the panels may be uniform in width, or in proportion to a cosine grid spacing. The cosine grid can be centered in order to provide denser paneling at both inboard and outboard blade segment edges, or biased only toward the inboard, or only toward the outboard edge. In practice the selection of a paneling scheme should allow denser definition in regions

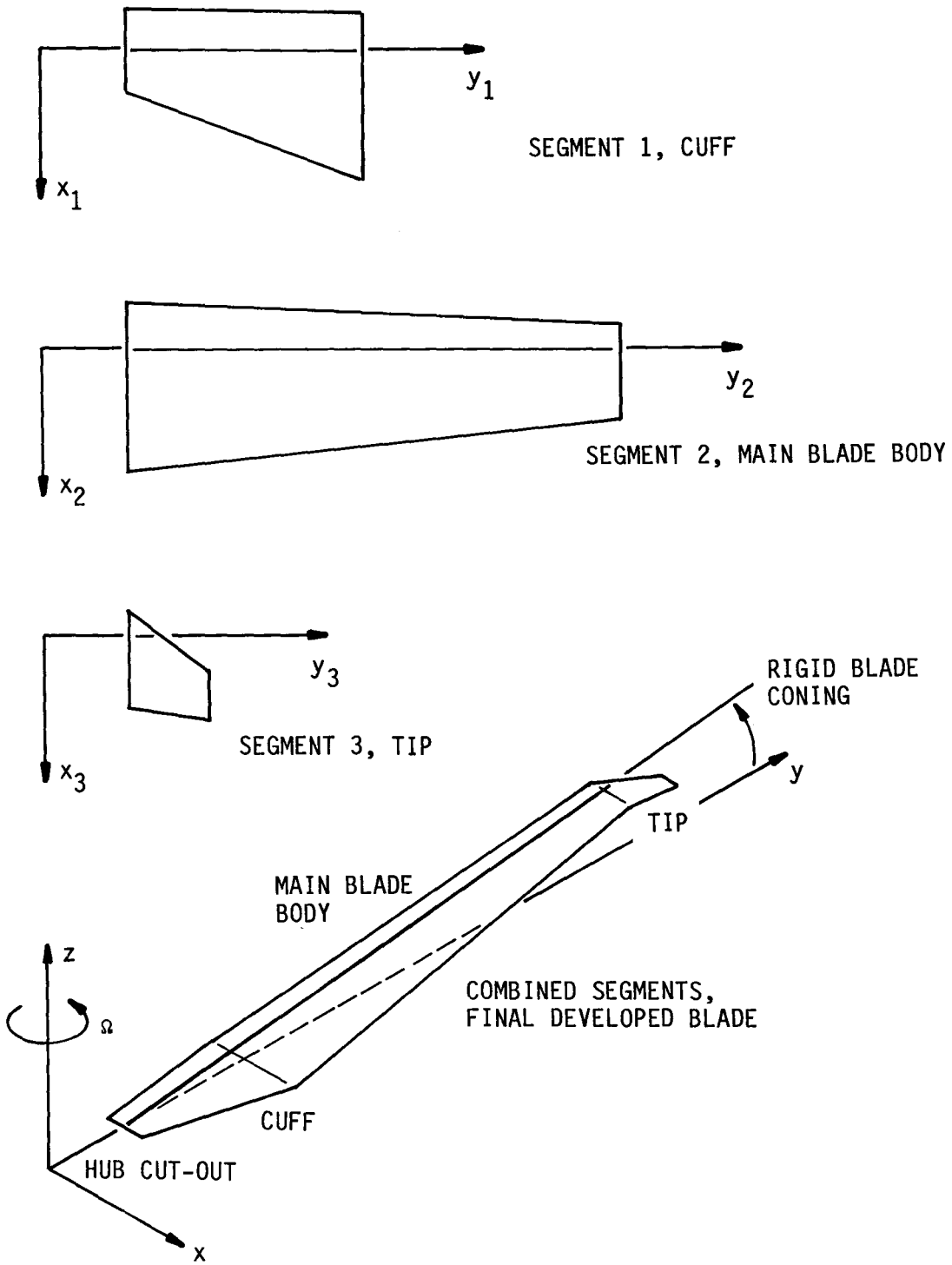


Figure 5. Development of Reference Blade Geometry From Segment Descriptions of Major Components.

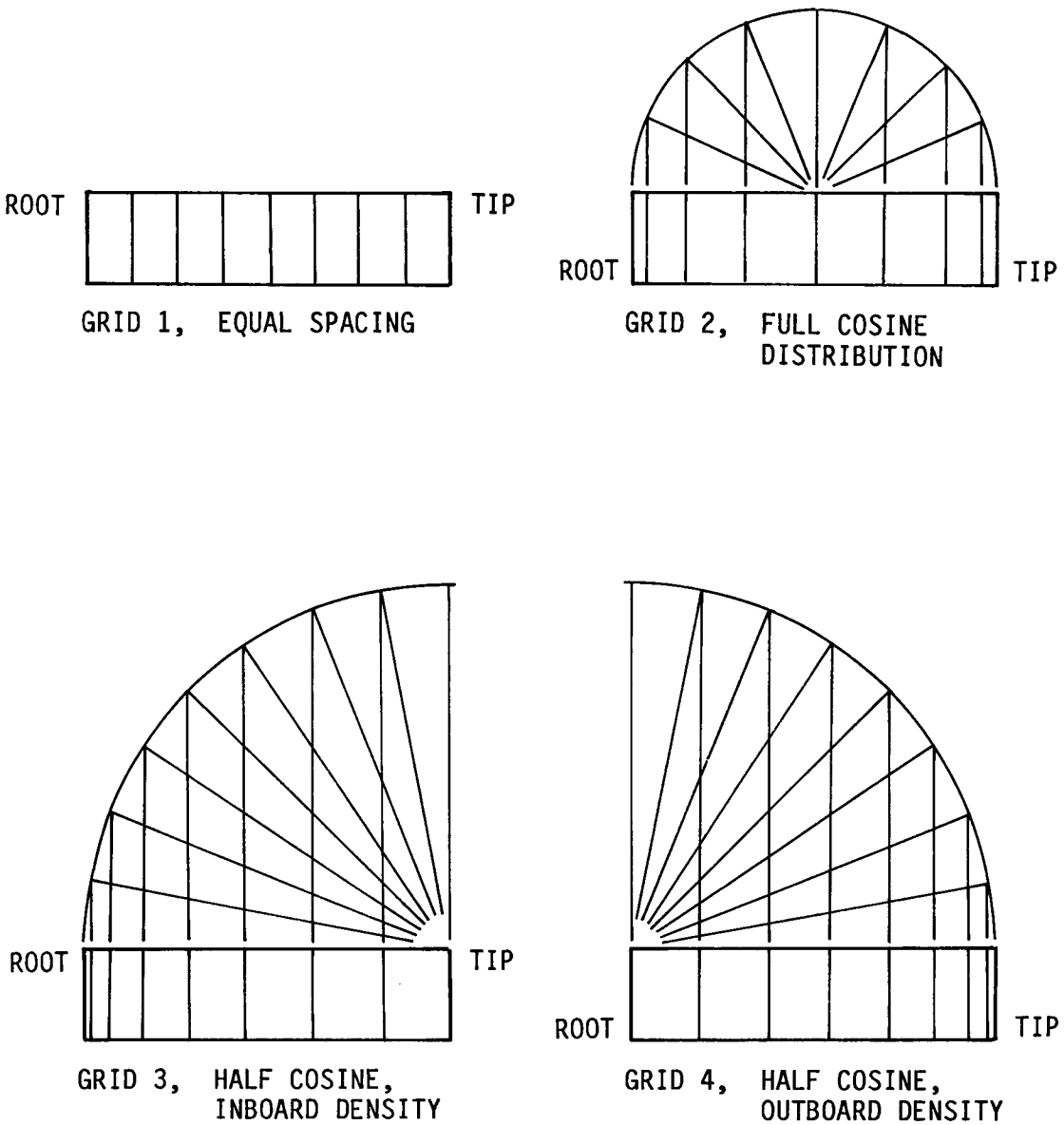


Figure 6. Spanwise Panel Distributions Corresponding to Blade Segment Grid Options.

which are anticipated to have larger gradients in surface loading. Grid generation is automated and requires no further user input.

In the chordwise direction the number of panels is constant for the entire blade. The chordwise arrangement is fixed and is based on a cosine grid biased toward the leading edge. This selection evolved from the work in References 2 and 3, as has the procedure for locating each panel's control, or normalwash collocation point. The numerical circulation solution is uniquely related to the location of these points and their optimized location greatly speeds solution convergence with respect to the number of panels employed. The chordwise locations are calculated from the known two-dimensional flat plate circulation distribution. This guarantees the correct self-induced velocity influence of each streamwise section of the lifting surface. Convergence of the solution then depends only upon a panel density which adequately accounts for the interference velocities from the remaining surface stations and the wake. The spanwise location of the collocation points is at the midspan of each panel. In general a 2 by 15 chordwise by spanwise grid of panels on the blade is sufficient.

Segment Description

The segment description begins with the number of spanwise panels, the type of grid desired, and the radial extent of the segment. Detailed segment geometry is supplied in a particular segments own axis system, although the same axis system may be used for all segments if convenient. Translations and rotations for each segment allow the segment to be placed in any desired location and orientation to form the developed blade. Thus the segment can be "built" in a convenient axis system and then placed in its proper location with respect to the rotor axis system.

Segment details are interpolated from piecewise linear radial distributions of chord, pitch increment due to twist, and planform offset (sweep). These distributions are supplied independently for each segment and are required to cover the radial extent of each. Also, the chord station $(x/c)_{ref}$ is specified which is used to initially align the chord

distribution along the segment span (y-axis). Figure 7 illustrates the effect of varying $(x/c)_{ref}$ for a tapered planform segment, The figure also shows how $(x/c)_{ref}$ is used in combination with an offset distribution to model a complex planform.

Airfoil Definition

The airfoil characteristics of a segment are defined by associating each span station of interest with tables of lift, drag and pitching moment coefficients which are bivarient with angle of attack and Reynold's number. A segment must have at least one airfoil defined. An airfoil may be defined over a range of the segment or at a point on the segment. As sketched in Figure 8, this allows regions of constant airfoil characteristics, transition regions bounded by defined airfoils, and butt joined airfoils. Within a transition region, the characteristics at an intermediate station are averaged from the bounding airfoils. In calculating the average, weighting is used in proportion to the distance to each bounding airfoil. Thus, at the exact center of a region a simple average results, and at the edge of a region the bounding airfoil is used exclusively.

REFERENCE BLADE DEVELOPMENT

The actual panel grid coordinates of the lifting surface are developed for a reference blade without displacements for collective pitch or flapping. The blade surface is then the warped surface of zero lift chordlines positioned by twist increments due to physical twist and the aerodynamic twist due to camber. This twist is applied about the segment y-axis. The zero lift line angle of attack is interpolated from the supplied airfoil tables. Rotations of the reference coordinates to pitch, azimuth, and flapping angles are subsequently performed as needed to create target and source blades for calculation of the AIC's.

The reference blade is built segment by segment. First, the untwisted planform is developed in the segment axis system from the chord and offset

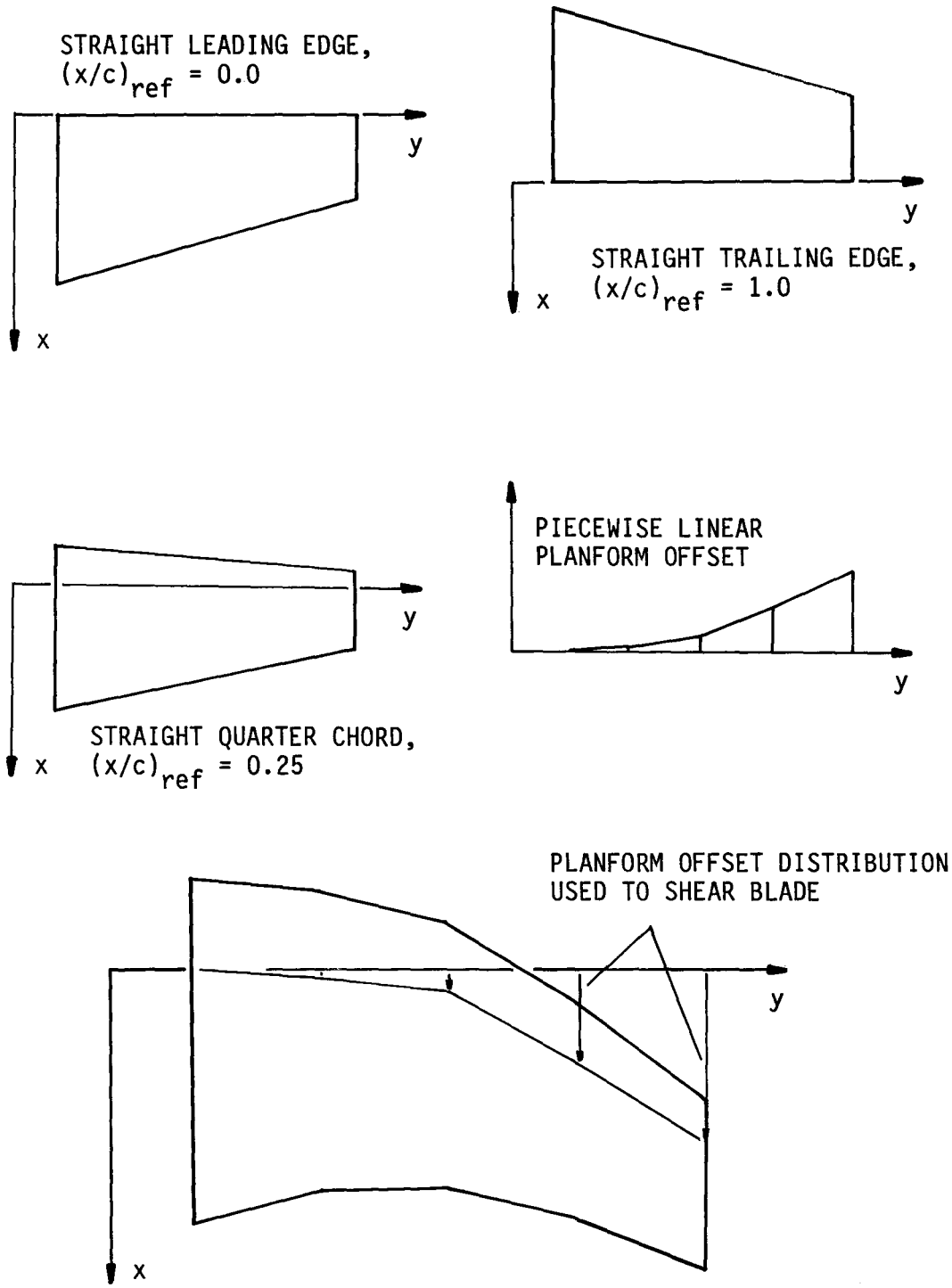


Figure 7. Modeling of Complex Planform With Segment Chord Reference Station $(x/c)_{ref}$ and Distributed Planform Offset.

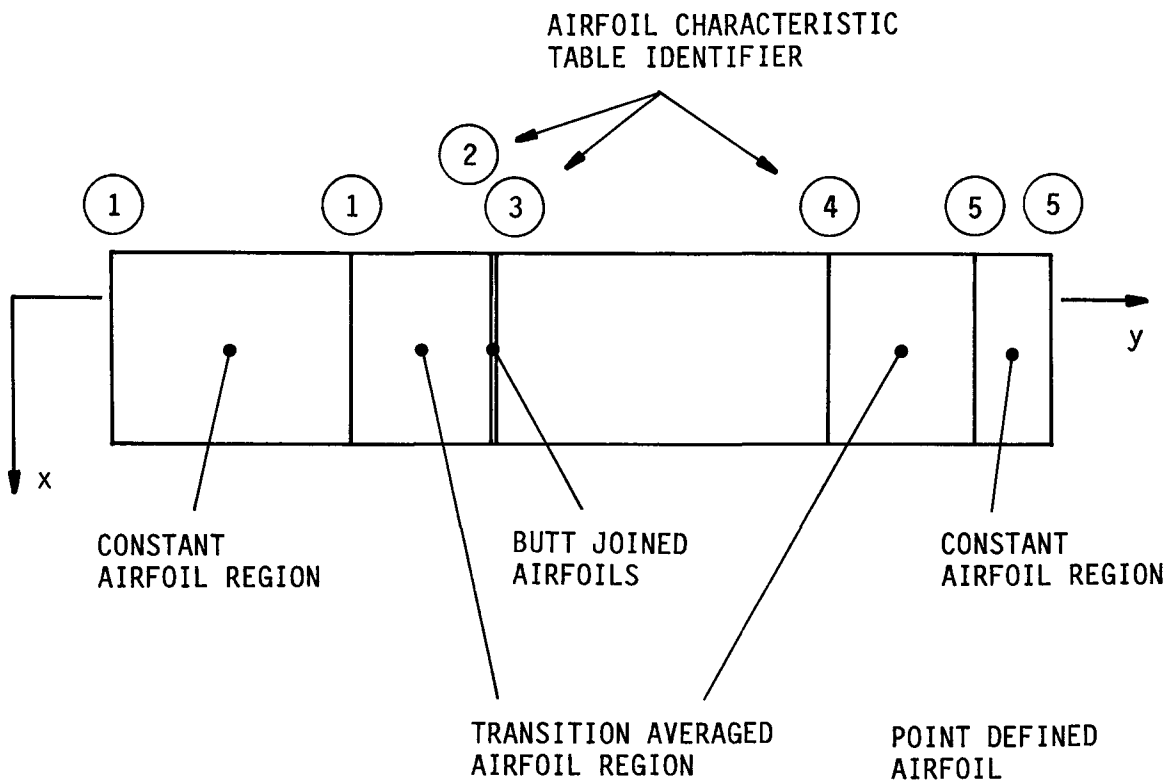


Figure 8. Association of Airfoil Characteristic Tables with Segment Spanwise Regions Allows Interpolation Between Tables.

distributions, and the reference chord station $(x/c)_{ref}$. Then the panel and collocation grid span stations are rotated about the segment y-axis through the combined physical and aerodynamic twist angle. Finally, y-, x-, and z- rotations about the segment axes, and x-, y-, z- translations are performed in the listed sequence to position the segment in the rotor axis system. For instance, if a common segment axis system is used then blade precone results from an x- rotation. If different systems are used for each segment then combinations of x- rotations and z- translations will provide the precone. As can be seen, several approaches through the blade development scheme can yield the same result. In all cases the user is cautioned to carefully check the output coordinate reports to verify the desired result.

GENERALIZED PRESCRIBED WAKE FOR AXIAL FLOW

The wake model for the wind turbine application is extended and simplified from those used for helicopter rotor hover analysis. The basic structural concept for the wake is that described by Gray in Reference 8, and reproduced here as Figure 9. Each blade's trailing wake is assumed to split into inboard and outboard sheets at the radial station corresponding to the blade's maximum bound circulation. The inboard sheet continues to settle in that form, and the outboard sheet quickly rolls to complete formation of a concentrated tip vortex. Although there is little experimental evidence of a concentrated root vortex, modeling capability is included in the wind turbine wake so that its analytical influence can be determined. This leads to the definition of an additional mid-sheet region, such that the inboard sheet can then roll-up to form a root vortex.

In hover these wake components are positioned through prescribed functions which give axial settling rates and radial contraction rates in terms of basic inflow parameters. Experimental measurements of the wake rates shows them to be primarily functions of the rotor thrust coefficient C_T

$$C_T = \frac{T}{\rho(\pi R^2)(\Omega R)^2} \tag{26}$$

which is related to the mean hover induced velocity v_h at the rotor disk as given by the actuator disk momentum theory,

$$v_h = (\Omega R)(C_T/2)^{1/2} \tag{27}$$

In describing the hovering wake the primary emphasis is on the tip vortex trajectory because the vortex remains in close proximity to the disk, and is the major source of induced velocity from the wake. Because of the proximity, the initial trajectory is also influenced by the local flow about the blade, and thus configuration parameters such as twist and taper

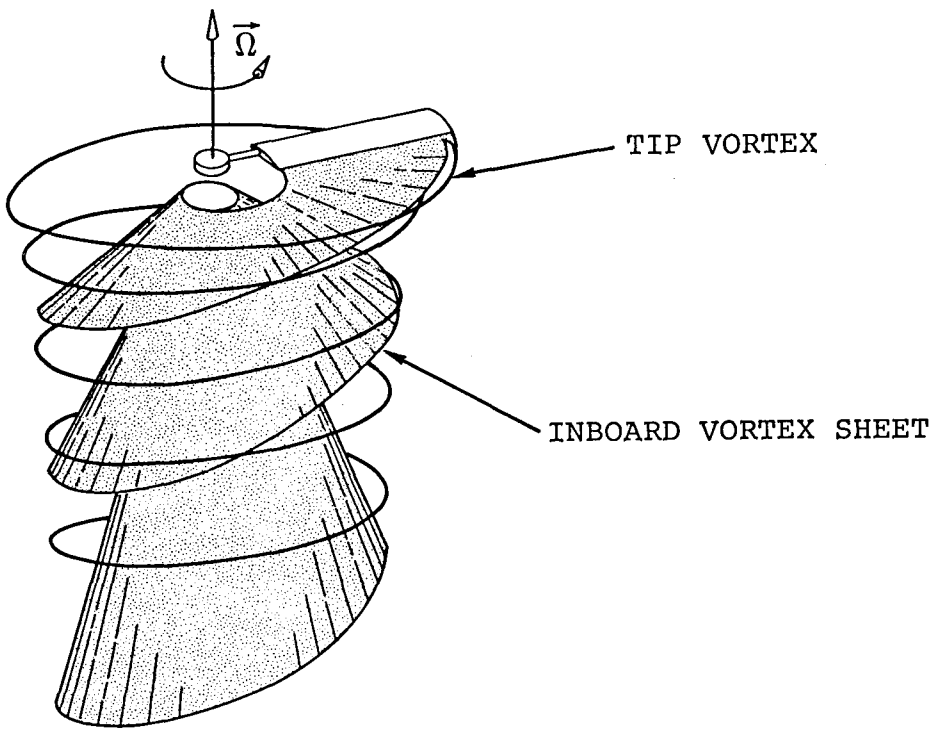


Figure 9. Gray's Concept of Hovering Rotor Wake Structure.

which influence the local aerodynamics. Figure 10, from Reference 3, shows measured axial and radial tip vortex trajectories of a hovering model rotor. The tip vortex initially settles at a near constant rate until passing beneath the following blade, at which time the additional induced velocity from the second blade's bound vorticity and wake increase the settling rate. The effects of subsequent blade passages appear minimal; thus, the secondary rate region is not significantly influenced by rotor details, and is dominated by the mean inflow of the rotor. The radial contraction behaves exponentially with increasing wake azimuth, reaching a minimum wake radius, after which the tip vortex trajectory usually becomes unstable as the vorticity diffuses.

This observed behavior is represented by parametric equations in ψ_w, the wake azimuth increment aft of the generating blade, of the form

$$z/R = k_z \psi_w \tag{28}$$

and

$$r/R = A+(1-A)e^{-\lambda\psi_w} \tag{29}$$

where k_z is the axial settling rate, λ is the contraction rate parameter, and A is the developed wake radius.

For the wind turbine case the trajectory rates will be augmented by the axial transport due to onset wind, which allows for some simplifications. First, note that the wake transport is attributable to induced and wind contributions. In the limit of very high wind speed there will be no induced effects because of the high pitch (settling rate) of the wake spiral. In this extreme all wake components, both sheet and concentrated vortices, will also experience the same transport, that due only to wind. It is further assumed then, that operating wind speeds are sufficiently high to require only mean induced effects to be included in describing wake trajectories, and that all wake component rates have the same induced contributions. The fact that the turbine wake will normally expand, rather than contract also supports this assumption since expansion results in increased displacement of the tip vortex trajectory away from the blade, even

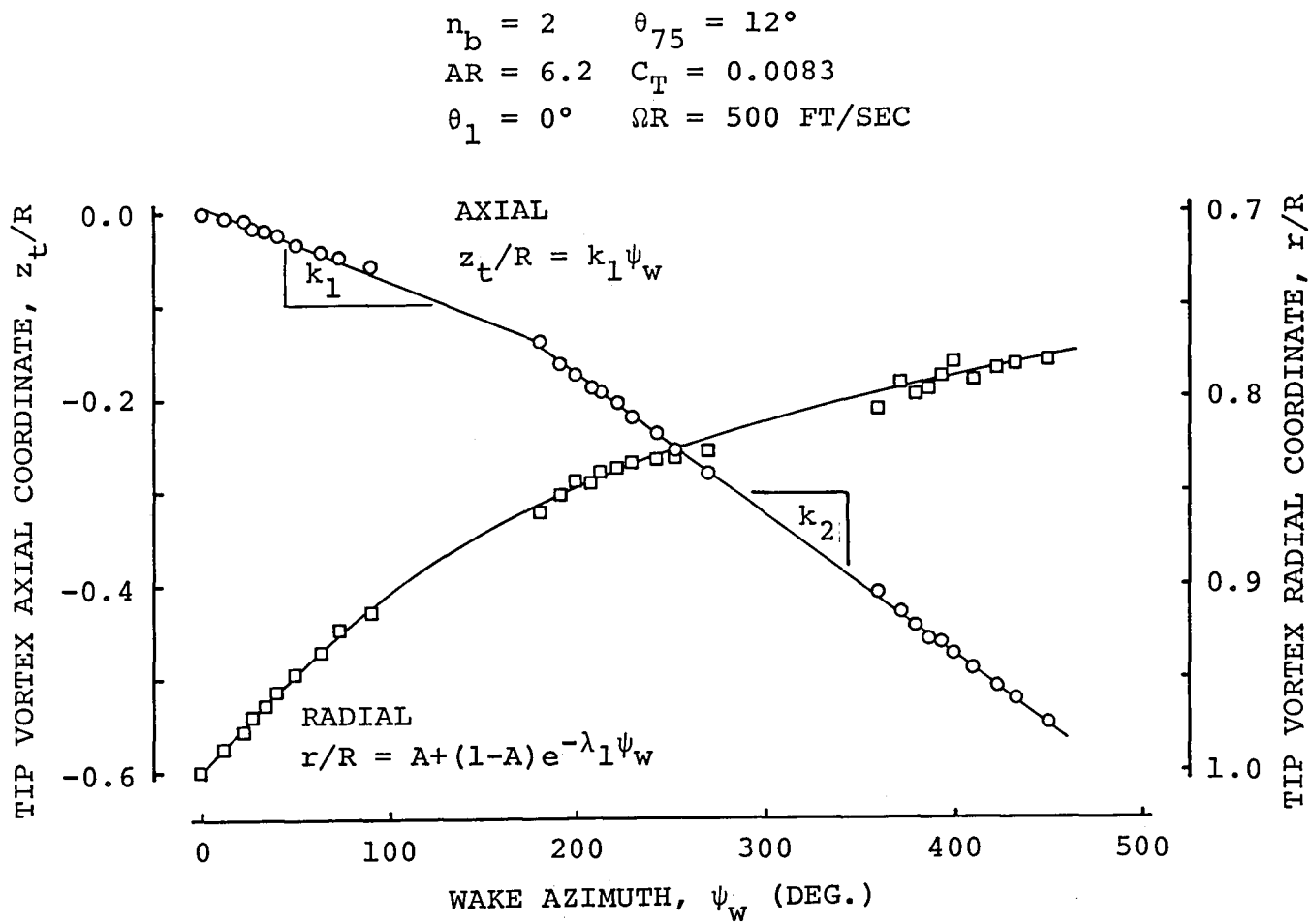


Figure 10. Model Rotor Tip Vortex Trajectory Measured From Schlieren Flow Visualization Photographs of Wake (Reference 3).

at lower wind speeds. Since only the wake position, and not the wake structure, is affected by these simplifications, then the model can be readily improved as better understanding or measurements are available.

The task is then to develop expressions for the turbine wake's axial and radial translation rates. Reference values in hover are first defined from the generalized expressions developed from measurements in Reference 3. The hover reference axial rate is taken as the secondary settling rate, and is given as

$$k_{z_h} = -(C_T)^{1/2} \tag{30}$$

The contraction rate parameter is generalized as

$$\lambda_h = 4.0(C_T)^{1/2} \tag{31}$$

But, defining contraction in terms of a displacement rate

$$k_{r_h} = \lambda_h/(k_{z_h}) \tag{32}$$

then the radial trajectory is more conveniently expressed as a function of the axial displacement,

$$r/R = A+(1-A)e^{-[k_{r_h}(z/R)]} \tag{33}$$

ROTOR WORK STATE EFFECT ON WAKE RATES

Equations (28) and (33) are the basic working expressions for generating the coordinates of the individual filaments of the wake, once the axial and radial displacement rates are modified to include the effects of wind speed. This wake rate modification is accomplished by forming an analogy between the rates and the average flow through the rotor as a function of the rotor work state.

The three basic rotor work (or flow) states are defined by the sense of the axial impressed velocity V_a (usually described as climb or descent rate in rotor terminology) through the rotor and by the presence or lack of an organized wake slipstream. A convenient format for describing the states is the plot of net flow through the rotor as a function of the axial impressed velocity. This format is generalized if all velocities are normalized by, or referred to, the absolute value in hover of the mean induced velocity from Equation (27). The referred axial, induced, and net velocities are then

$$\overline{V}_a = V_a/v_h \tag{34}$$

$$\overline{v} = v/v_h \tag{35}$$

and,

$$\overline{V}_{net} = \overline{V}_a + \overline{v} \tag{36}$$

The normal state extends from hover through climb, $\overline{V}_a \leq 0$. The impressed velocity and induced velocity are both downward through the rotor and $\overline{V}_{net} \leq \overline{v}$. From hover into descent the rotor enters the vortex ring state in which the induced velocity is opposed by the descent speed. At the center of this state the apparent net flow through the rotor vanishes, and the persistence of streamline flow through the rotor diminishes until a definite slipstream in the vicinity of the rotor fails to exist. Characteristically, a highly unstable air-body of recirculating flow develops about the rotor, from which the flow state is named. The limits of the vortex ring state are $0 \leq \overline{V}_a \leq 2$. At higher descent speeds the windmill brake state is entered and streamline flow reorganizes. This, of course, is the work state of primary interest for wind turbine analysis. In the normal and windmill brake states, because of the existence of a continuous streamtube, the disk momentum theory is applicable, and gives the mean induced velocity for the two states respectively as

$$\overline{v} = -\overline{V}_a/2 - [\ (\overline{V}_a/2)^2 + 1\ ]^{1/2} \tag{37}$$

and

$$\overline{v} = -\overline{V}_a/2 + [\ (\overline{V}_a/2)^2 - 1\ ]^{1/2} \tag{38}$$

In the vortex ring state, momentum theory fails to give observed results for the induced velocity because of the breakdown of the streamline structure of the flow. However, it does give approximate trends at the initial entry from hover at very low descent speeds. The complete description of the vortex ring region must therefore be accomplished with empirical relationships implied by the apparent mean induced velocity required to match experimental measurements of rotor thrust or torque to corresponding values calculated from momentum theory.

In wind turbine analyses, the most widely used empirical relationship between induced and axial stream velocity in the vortex ring state is that due to Glauert from his 1926 work (Reference 12). More recent (1951) measurements on rotors in descent made by Castles and Gray in Reference 7, show considerable discrepancy in comparison to Glauert's results. At a referred descent velocity of about 1.5, the turbulent wake begins its transition toward the organized streamlines of the windmill brake state. In this region the Glauert empirical curve yields about one-half the induced velocity as measured by Castles and Gray. Their data also agrees with helicopter measurements while in ideal autorotation, and the authors suggest that Glauert's data is in error because of unaccounted for changes in blade angle of his solid brass model rotor when under load. As the Glauert empirical model is in popular use in momentum analyses for wind turbines, this discrepancy could yield misleading results at low wind speeds.

For the present application, an empirical model based on Castles and Gray is preferred. Additionally, the work of Drees et al in Reference 13, supports that of Castles and Gray. This Reference also presents smoke flow visualization studies of the entire axial flow regime and provides a complete discussion of the flowfield behavior in the vortex ring state.

With a degree of idealization the apparent induced velocity in the vortex ring state can be simply represented by two linear functions as

$$\bar{v} = -\bar{V}_a - 1, \qquad 0 \leq \bar{V}_a \leq 1.5 \tag{39}$$

$$\bar{v} = 3\bar{V}_a - 7, \qquad \bar{V}_a \geq 1.5 \tag{40}$$

These relationships are plotted in the work state velocity diagram presented as Figure 11. A very interesting result seen in this format is that based on the Castles and Gray measurements, the net flow through the rotor from hover to $\overline{V}_a = 1.5$ is nominally constant and equal in magnitude to the induced velocity in hover.

With the preceding discussion in mind the behavior of the wake parameters in axial flow is now developed. It is first assumed that the net settling rate in the normal and windmill brake states is composed of an induced component and one due to the impressed axial velocity transport,

$$k_z = k_a + k_i \tag{41}$$

where

$$k_a = V_a/(\Omega R) \tag{42}$$

and where k_i is proportional to the induced velocity. Proceeding as before with the velocities, the settling rate components are referred to the reference value from hover

$$\begin{aligned} k_{z_h} &= |-(C_T)^{1/2}| \\ &= |\sqrt{2}\,[v_h/(\Omega R)]| \end{aligned} \tag{43}$$

yielding

$$\overline{k}_z = k_z/k_{z_h} \tag{44}$$

$$\overline{k}_a = \overline{V}_a/\sqrt{2} \tag{45}$$

The functionality of k_i is determined now by recognizing that the pitch of the wake vortex helix is given by the settling rate and that the induced velocity varies inversely with the pitch. Therefore

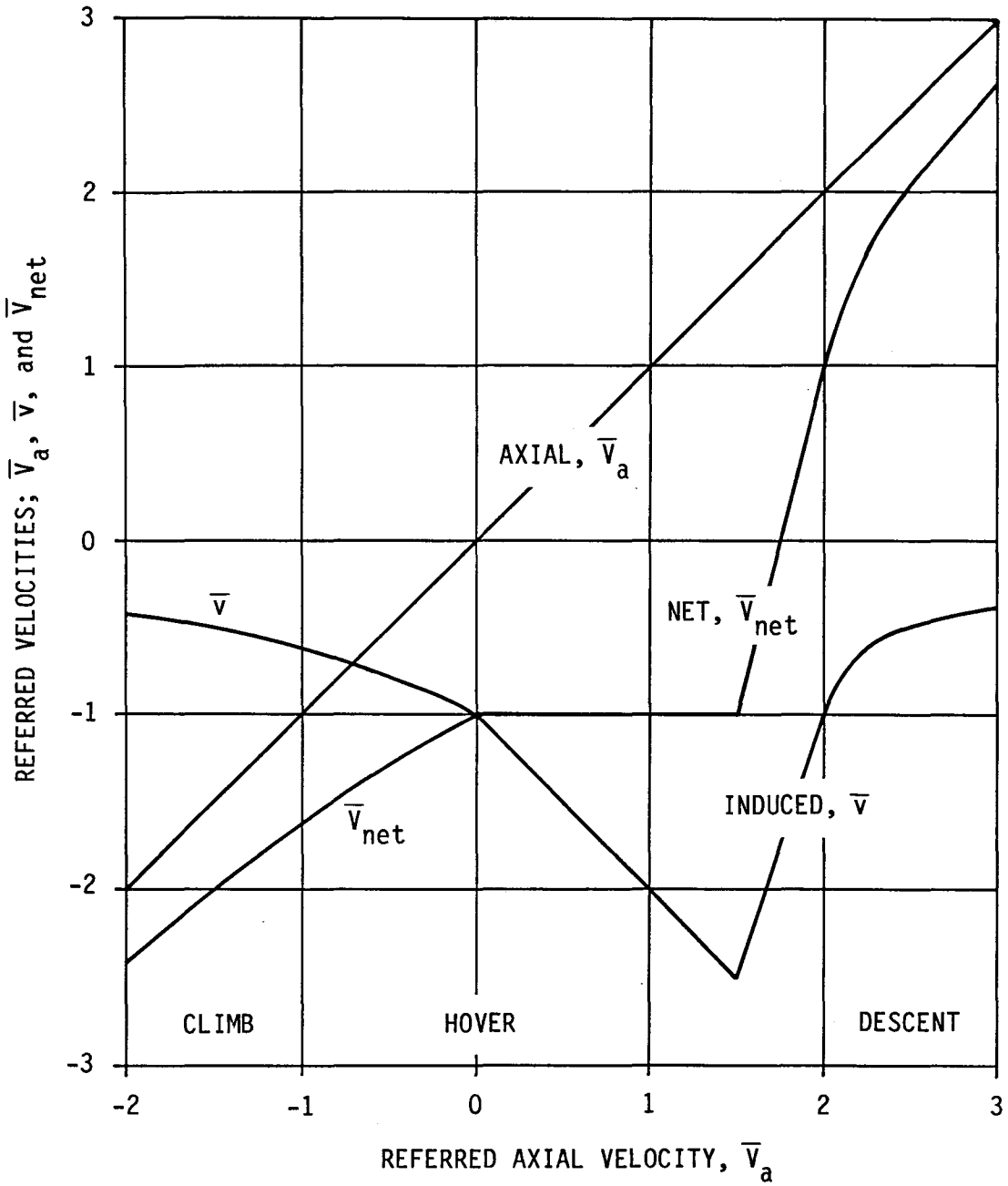


Figure 11. Rotor Work State Velocity Relationships in Axial Flow.

$$\bar{k}_z = \bar{k}_a + [\text{constant}]/\bar{k}_z \tag{46}$$

For the normal work state the constant is determined by matching the known conditions at hover, $\bar{k}_a = 0$ and $\bar{k}_z = -1$, giving the constant a value of 1. Then Equation (46) is solved to give the final expression for the settling rate for the normal work state,

$$\bar{k}_z = (\bar{k}_a/2) - [(\bar{k}_a/2)^2 + 1]^{1/2} \tag{47}$$

In the vortex ring state as previously noted, the organized streamline flow transitions to a complex large scale circulation about the rotor, and therefore the wake spiral most probably also becomes disorganized. However, in the prescribed wake model it is assumed that the integrity of the wake spiral remains, and thus its position should nominally represent the overall circulation of the flow. This is accomplished by recognizing at hover that the settling rate is purely due to induced velocity, but that the induced velocity is also the net flow through the rotor at this key point. It is further assumed that in the vortex ring state the settling rate continues to be proportional to the net flow. From Equations (39) and (40) the settling rate in the vortex ring state is simply given by

$$\bar{k}_z = \bar{V}_a + \bar{v} \tag{48}$$

$$\bar{k}_z = -1, \qquad 0 \leq \bar{V}_a \leq 1.5 \tag{49}$$

$$\bar{k}_z = 4\bar{V}_a - 7, \qquad 1.5 \leq \bar{V}_a \leq 2 \tag{50}$$

In addition to matching the end conditions at the axial velocity limits of the vortex ring state, note that the settling rate will also reduce to zero at $\bar{V}_a = 1.75$, (the wake will remain inplane with the rotor disk) and change sign (the wake will move from beneath the rotor, the upwind side, to the downwind side). This point corresponds to the ideal autorotation vertical descent speed at which the rotor flowfield is analogous to that of a solid disk; that is, there is no net axial flow. Thus the present modeling

approach at least intuitively models the accumulation of the wake into a vortex ring and the subsequent transition toward the windmill brake state.

Once into the windmill brake state the settling rate is modeled as in the normal state. Determining the constant in Equation (46) now at $\overline{V}_a = 2$ or $\overline{k}_a = 2$, and $\overline{k}_z = 1$, then the constant is $(1-\sqrt{2})$ and the settling rate is

$$\overline{k}_z = (\overline{k}_a/2) + [(\overline{k}_a/2)^2 + (1-\sqrt{2})]^{1/2} \qquad (51)$$

Figure 12 presents the work state diagram in terms of the settling rate as described by the several equations above, and the analogy to the velocity work state diagram is clearly seen.

The point at which the wake migrates from upwind to downwind of the rotor disk is also taken as the nominal condition when the radial character of the wake changes from contraction to expansion. Maintaining the form of Equation (33) for the wake radial coordinate throughout the axial regime,

$$r/R = A+(1-A)e^{-[k_r(z/R)]} \qquad (52)$$

Then the developed wake radius A and the rate parameter k_r variation with axial velocity must be determined. The reference value in hover of A = 0.78 is again taken from the experimental measurements of Reference 3. In the normal work state the wake will contract, but at a rate decreasing with axial velocity to a negligible value in the high speed limit such that contraction is also negligible, even if the developed wake radius remains at the hover value. Similarly, the developed radius of the expanding wake is of little importance in the other extreme of axial velocity, so that the developed wake expanded radius is only important at low speed. Without guiding experimental data, the value is set by assuming the same expansion increment as for contraction, thus giving as a function of the sign of the settling rate

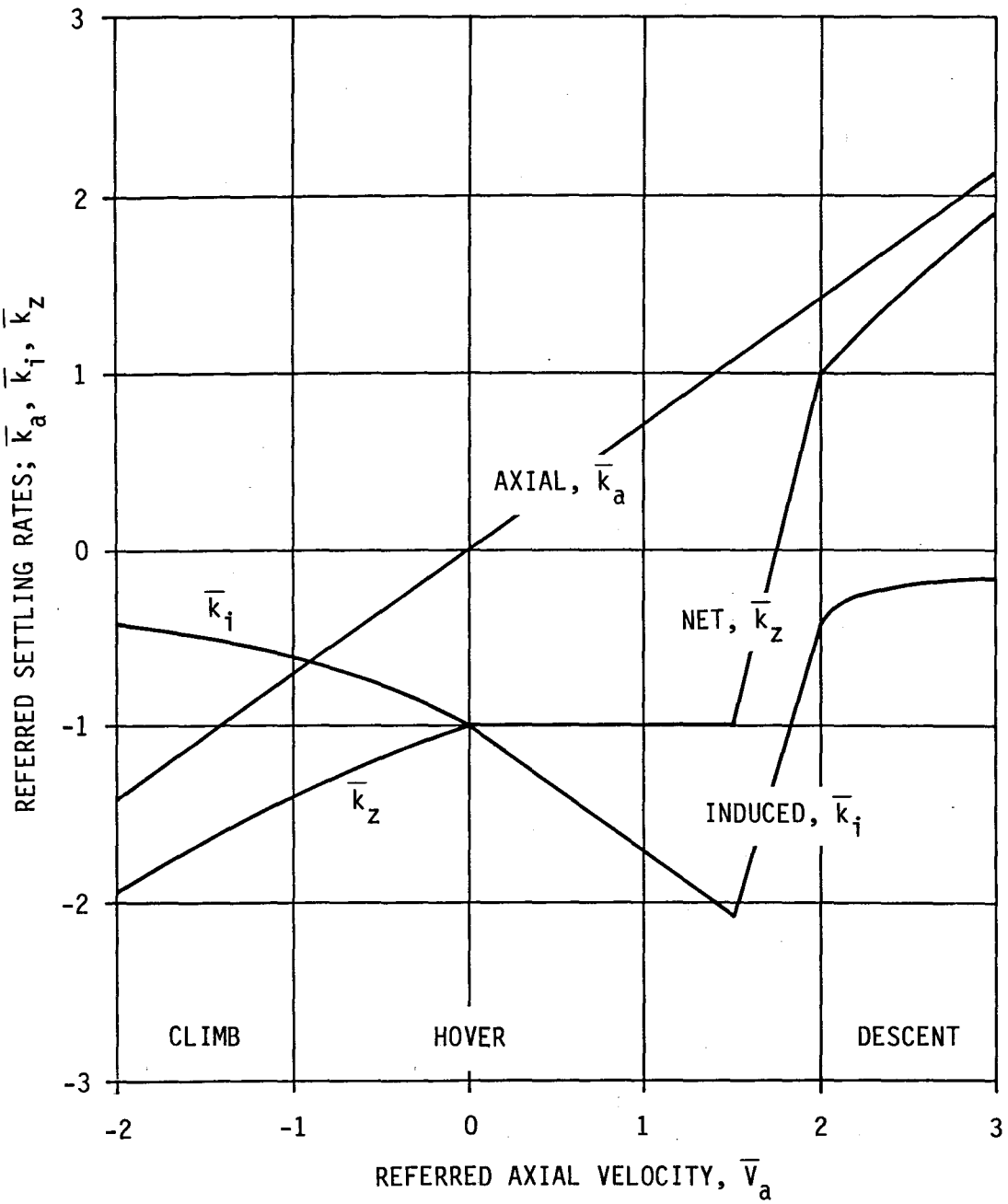


Figure 12. Rotor Work State Settling Rate Relationships in Axial Flow.

$$A = 1 + 0.22[\mathrm{sign}(k_z)] \tag{53}$$

The rate parameter is assumed to be inversely proportional to the wake pitch which sets the radial induced velocities. This also provides the proper vanishing rate at the high speed axial velocity limits. With consideration for the required sign then

$$k_r = |k_{r_h}/\bar{k}_z|[\mathrm{sign}(k_z)] \tag{54}$$

Equations (28) and (33) can now be used to generate the cylindrical coordinates of the wake as a function of the wake azimuth. The wake is composed of a number of trailers, each with an origin at the intersection of a blade surface panel boundary with the blade trailing edge, and the generating equations are adjusted with the appropriate offsets to match the origin (r_{te}, z_{te}) of a particular trailer,

$$z/R = k_z(\psi_w - \psi_{te}) + (z_{te}/R) \tag{55}$$

$$r/R = (r_{te}/R)\{A+(1-A)e^{-[k_{r_h}(z-z_{te})/R]}\} \tag{56}$$

ROLLUP OF CONCENTRATED VORTICES

The implementation of the prescribed wake model is designed to allow the inboard and outboard sheets to roll up into concentrated root and tip vortices respectively. The rollup of the two vortices may be selected independently or in combination, and a portion of the wake called the mid-span sheet can be included without any rollup if desired. Also, the rollup mechanism can be suppressed altogether, fixed to include a series of specified trailers, or for the tip vortex allowed to vary through iteration to be compatible with the calculated radial bound circulation distribution. Compatibility is interpreted to mean that the inboard edge of the outboard sheet coincides with the radial position of the outermost maximum in the bound circulation distribution. The generality in selection of the rollup

is provided to allow exploration of the importance of the several wake structure elements and to accommodate future refinements of the wake model's structure as data and better understanding of the real wake become available.

Figure 13, shows the overall scheme of the wake grid and rollup options. The near wake extends from the blade trailing edge through one revolution. In the near wake the number of azimuth increments is increased toward the blade to improve the fidelity of the discrete, piecewise linear wake elements to the actual curved trajectory of the wake filament trailers. The number of azimuth increments per revolution in the nonuniform grid of the near wake is twice that specified in the uniform grid region that follows and continues to the truncation point. At the truncation of the uniform grid the wake can optionally be continued with the semi-infinite vortex cylinder described in a previous section.

The optional rollup process is described first by specifying the bounding indices of the trailers on the edges of a sheet that will develop into a concentrated vortex. Then the azimuth increment is specified over which the sheet will form into the vortex. The index of a trailer corresponds to the blade surface panel grid index from which the trailer emanates. As shown in Figure 14, the grid indices are a function of the number of panels spanwise within a blade segment, and the number of blade segments.

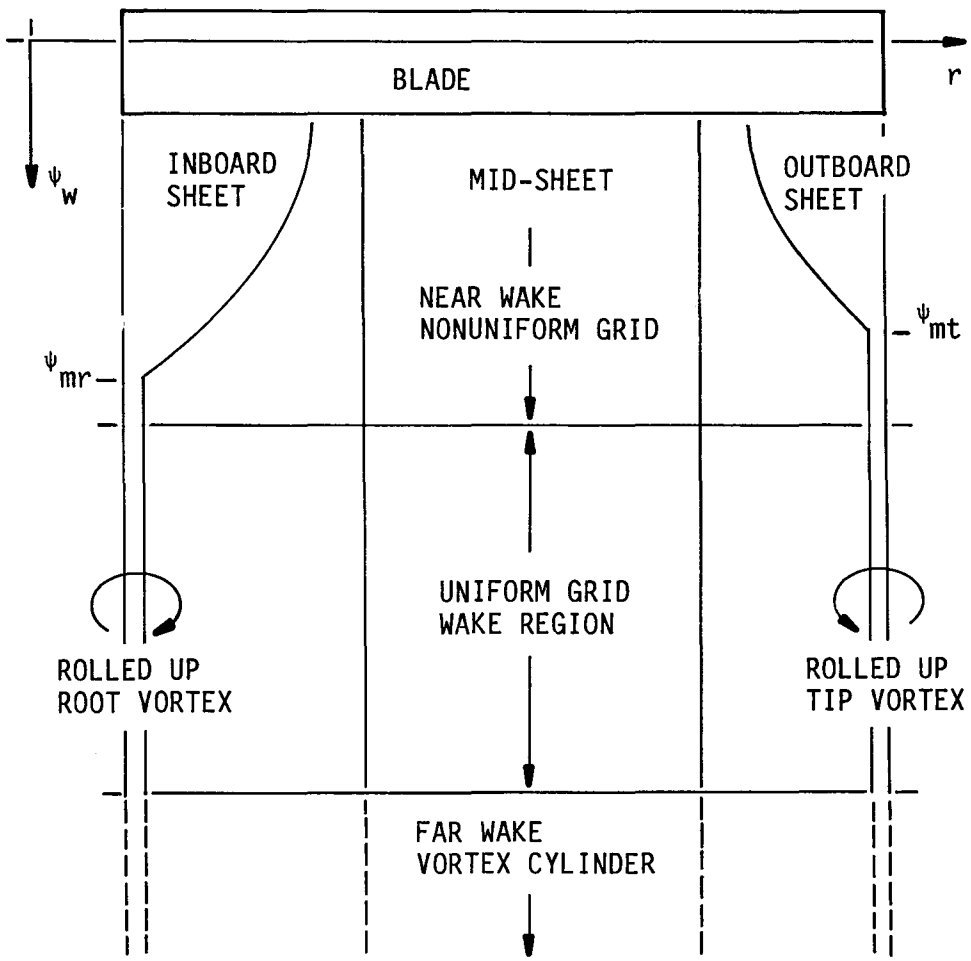


Figure 13. Schematic of Wake Grid Regions, Sheet Definitions, and Rollup Options.

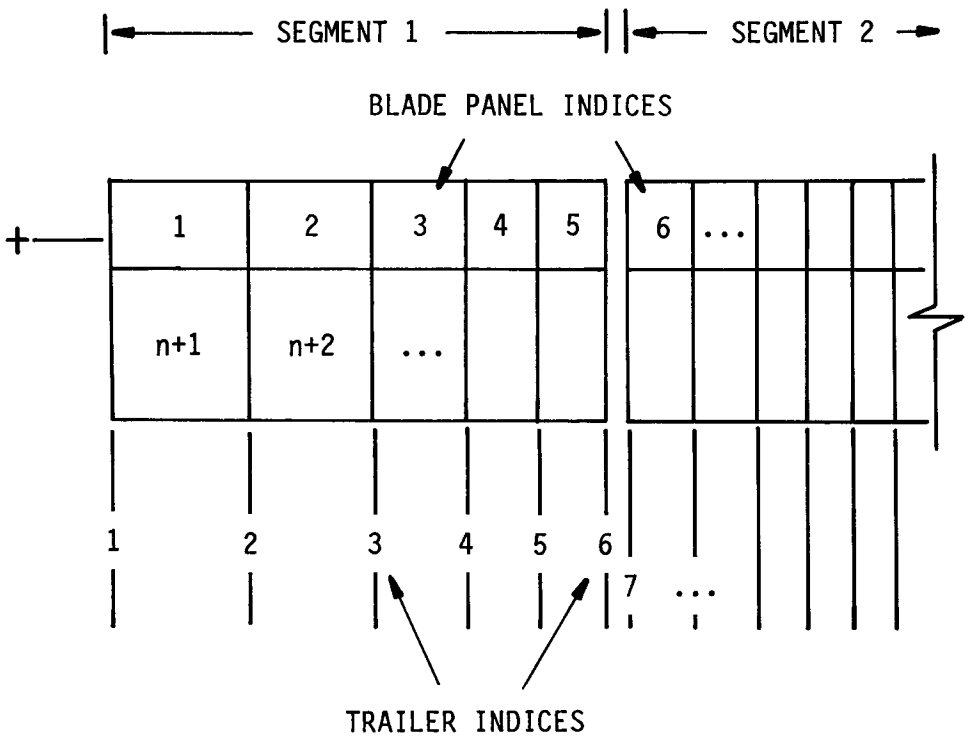


Figure 14. Panel, Panel Boundary Grid, and Wake Trailer Index Assignments.

PERFORMANCE CALCULATION

With the general characteristics of the lifting surface theory and prescribed wake now defined, the overall solution procedure can be outlined. The principal steps in obtaining a converged solution of the detailed aerodynamics and integral performance are described in the flow chart of Figure 15. The major iteration cycle of the problem is to seek a thrust level and wake geometry which are mutually compatible at the specified blade pitch. Secondary iterations are performed to insure correspondence of the sectional effective linear lift curve slope with the potential flow circulation solution, and to develop the outboard sheet boundary and tip vortex rollup if optioned.

For given blade and rotor geometry, and with an iterative estimate of the wake settling and contraction rates, the discretized panel representation of the blade surfaces and wakes is developed in cartesian coordinates. Then an aerodynamic influence coefficient matrix is assembled from separate calculations of the component coefficients due to each blade's surface and wake. Gauss-Seidel iteration is used to obtain the circulation solution from the resulting linear equation system.

The blade element effective angle of attack is calculated from the circulation solution at each chordwise column of surface panels. The effective angle is related to the bound circulation, that is the value of distributed circulation at the trailing edge Γ_{te}, through the Kutta-Joukowski theorem,

$$\sin(\alpha_e) = \frac{2\Gamma_{te}}{a_e V_r c \cos(\Lambda)} \tag{55}$$

where c is the blade element chord and

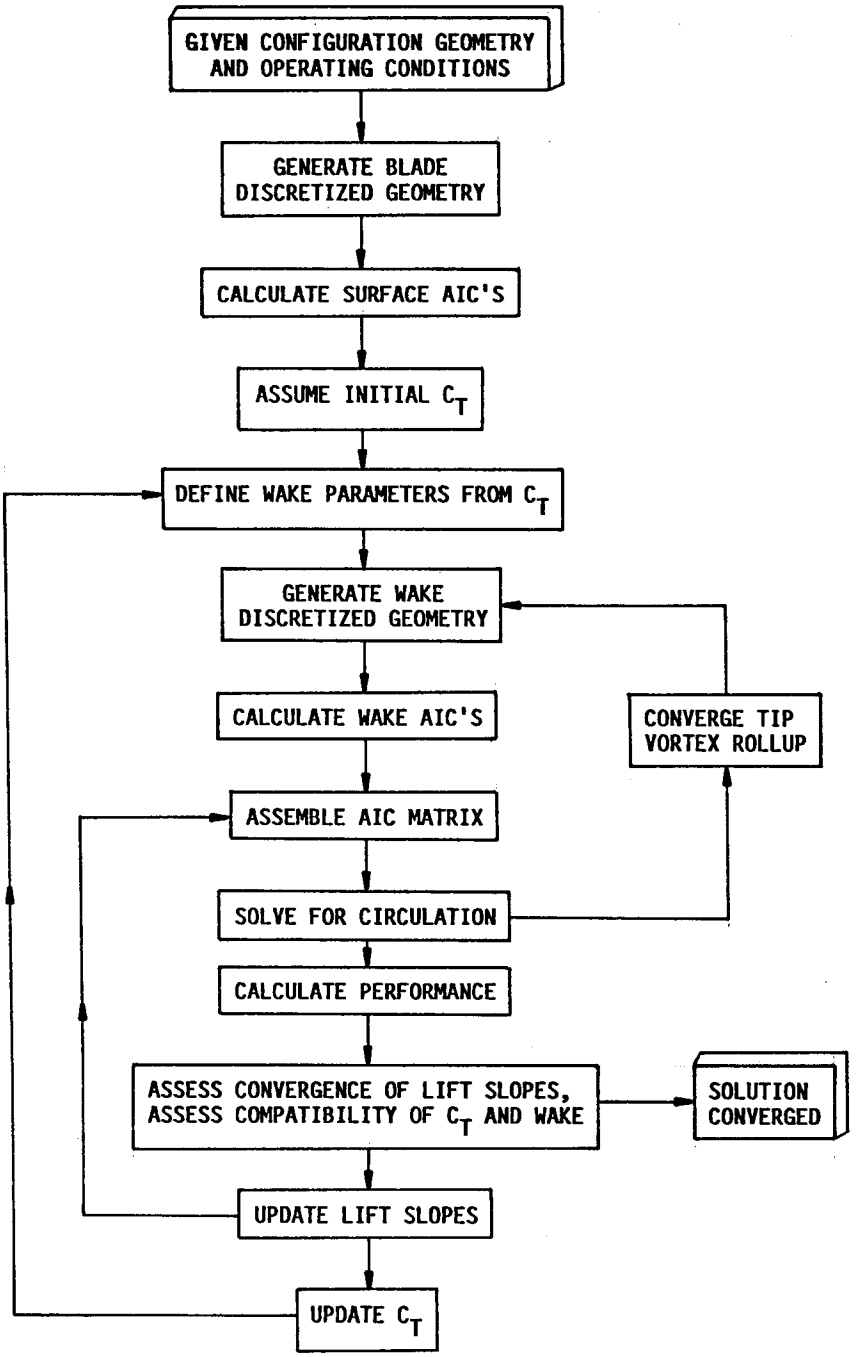


Figure 15. Flow Chart of Major Steps in Solution Procedure.

$$a_e = \frac{c_l}{\sin(\alpha_e)} \tag{56}$$

is the effective linear lift curve slope available from a previous iteration. The blade element relative velocity V_r contains the full freestream vector plus the induced swirl and downwash. It is calculated from the AIC's and the now known circulation, and the single blade element value of V_r is taken as the chordwise average value. The aerodynamic sweep angle Λ is the angle between the V_r vector and the blade element quarter chord. Also, the use of the large angle form of Equations (55) and (56) is required because of the very high angles that a wind turbine can encounter at high wind speeds when operating near peak power output.

Below stall, the total inflow angle is deduced from the difference between the geometric angle adjusted for zero lift line offset,

$$\alpha_n = \alpha_e - \alpha_g + \alpha_0 \tag{57}$$

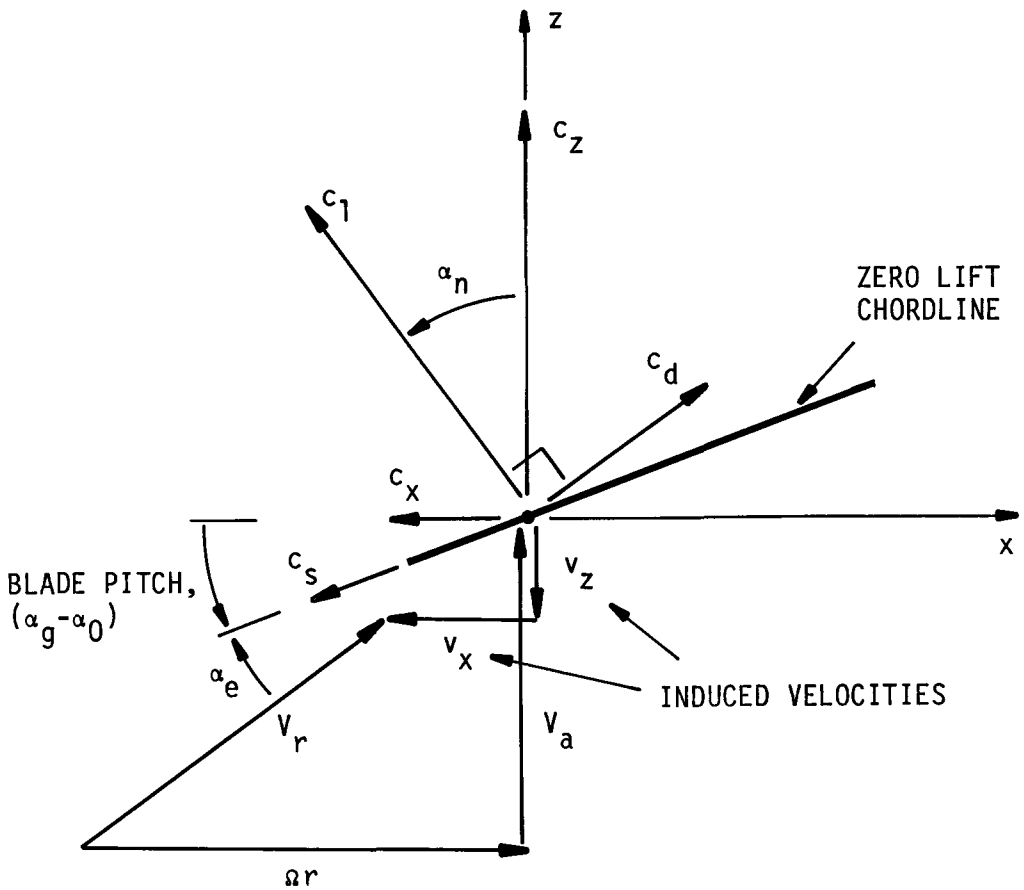
and the induced angle is then the difference between the total inflow angle and that due only to the axial freestream,

$$\alpha_i = \alpha_n - \alpha_w \tag{58}$$

The blade element airfoil lift and drag coefficients, c_l and c_d, are obtained from bivarient tables of two-dimensional coefficients as functions of angle of attack and Reynolds number. The coefficients are also corrected for sweep using infinite yawed wing theory and the blade element sweep angle Λ. The aerodynamic coefficients are then resolved as shown in Figure 16, into element z- and x-forces (thrust and torque forces) using the dynamic pressure based on V_r and the relations

$$c_z = c_l \cos(\alpha_n) + c_d \sin(\alpha_n) - c_s \sin(\alpha_g - \alpha_0) \tag{59}$$

$$c_x = c_l \sin(\alpha_n) + c_d \cos(\alpha_n) + c_s \cos(\alpha_g - \alpha_0) \tag{60}$$



NOTE: BLADE PITCH, INDUCED x,z-VELOCITIES, SUCTION FORCE INCREMENT, AND x-FORCE ARE NORMALLY NEGATIVE AS SHOWN

Figure 16. Velocity, Force Coefficient, and Angle of Attack Resolution at a Blade Element.

The element forces and torques are then integrated radially to give the final configuration performance. The coefficient c_s is the increment in leading edge suction force due to three-dimensional effects such as finite planform,

$$c_s = -\left(\frac{2\Gamma_{te}}{V_r\ c}\right)^2 \frac{(g_0{}^2-1)}{2\ \cos(\Lambda_{le})} \tag{61}$$

The two-dimensional portion of the suction force is assumed to be contained in the tabular airfoil coefficients. The suction force is proportional to the coefficient g_0 which is derived from the deviation near the leading edge of the chordwise circulation distribution from the two-dimensional flat plate value. The force acts in the chord plane, normal to the leading edge, and so is resolved through the leading edge sweep angle Λ_{le}. Further detail of the suction force calculation may be found in Reference 1.

In the pre-stall range of angles of attack the use of the effective linear lift curve slope causes the potential flow circulation response to match that of the airfoil characteristics. After stall occurs the procedure will still continue to match the calculated circulation to the stalled airfoil behavior. In this case the process is of course only simulating a very complex aerodynamic environment by forcing the blade element to stall in the sense of a two-dimensional airfoil. Furthermore, it is no longer clear how to relate any calculated leading edge suction increment to the fully separated real flow, or how to properly calculate the induced angle to resolve the element aerodynamic coefficients, which at high angles are themselves simply resolved from the normal pressure force. It is reasoned that the local flow should be dominated by the separation and so only the suction force already contained in the two-dimensional airfoil characteristics is considered post-stall. The effective angle of attack is continued to be calculated from Equation (55), but the total inflow and induced angles are now implied directly from the average chordwise velocity vector at the blade element. This approach is believed to more correctly resolve the separated flow force coefficients.

SIMULATION OF POST STALL AIRFOIL CHARACTERISTICS

Wind turbine aerodynamic calculations and performance estimates are additionally complicated by the fact that the blades are partially stalled for all wind conditions. As any practical analysis must rely on airfoil characteristics included through a blade element representation of the local performance quantities, considerable attention is given to modeling high angle of attack airfoil behavior. Most approaches are based on stalled flat plate aerodynamics with corrections to address suspected three-dimensionality of the stall phenomena.

Assuming the availability of reliable measurements up through the stall angle associated with maximum lift coefficient, and assuming that the fully separated flow is adequately modeled by the flat plate theory, then the transition region from stall onset to the point that the flow recovers to the flat plate behavior is of critical interest. This transition generally occurs in the angle of attack range from about 15 to 30 degrees, based on observation of the very limited amount of detailed experimental data in this range. Airfoil data is difficult to measure in this transition because both surface pressure and wake rake measurements become unreliable after stall, especially for the measurement of drag. Once the flow is fully separated, and assuming that two-dimensionality of the flow is maintained, then integration of surface pressure can yield reliable results.

The calculation of wind turbine peak power is very sensitive to the treatment of stall transition airfoil characteristics, and as yet there does not appear to be a completely satisfactory and validated model. In the implementation of the LSWT analysis several post-stall models are available in addition to directly using tables at high angles. These models include the Aerovironment, Inc. stalled flat plate model from Reference 6, the synthesis method developed by Viterna in Reference 14, and a method developed as part of the present work which allows more flexibility in selecting parameters to explore the sensitivity of the model. The Aerovironment method is included for completeness, but will not be

described in detail because it requires the transition region characteristics to be part of the supplied table.

VITERNA STALL MODEL

This method synthesizes the post stall lift and drag characteristics based on idealizing the stall such that the calculated torque at high angles of attack is constant. This requirement is based on the observation for some large scale turbines such as the NASA Mod-0, that the peak power remains relatively flat with increased wind speed. It should be noted that the derivation is strictly valid only for untwisted blades operating at flat pitch. The result is a set of analytical expressions giving c_l and c_d for all angles above that for stall.

A parameter in the method is the drag coefficient at $\alpha = 90$ degrees,

$$c_{d_{max}} = 1.11 + 0.018(AR) \tag{62}$$

The variation of the maximum drag with aspect ratio AR is a correction for three-dimensionality based on the data presented in Reference 15, for fully stalled finite span flat plates. With the subscripts l, d, and s denoting respectively lift, drag, and the value of a parameter at stall, the Viterna model is described by the equation set

$$B_1 = c_{d_{max}}$$

$$B_2 = [c_{d_s} - B_1 \sin^2(\alpha_{s_d})]/\cos(\alpha_{s_d})$$

$$A_1 = B_1/2$$

$$A_2 = [c_{l_s} - c_{d_{max}} \sin(\alpha_{s_l})\cos(\alpha_{s_l})] \sin(\alpha_{s_l})/\cos^2(\alpha_{s_l})$$

$$c_l = 2A_1\sin(\alpha)\cos(\alpha) + A_2\cos^2(\alpha)/\sin(\alpha)$$
$$c_d = B_1\sin^2(\alpha) + B_2\cos(\alpha) \qquad (63)$$

In implementation the stall angles are the maximum angles contained in the airfoil tables of lift and drag coefficients, and the aspect ratio is based on the blade chord at 0.75 radius. The use of aspect ratio in this manner is really equivalent to the assumption that the entire blade stalls at once. If the radial extent of the stall were only partial then the degree of relief would intuitively be a function of a stalled elements proximity to a free edge, if in fact the stall region included either the blade root or tip. If the region were bounded by attached flows then it might be maintained more nearly as two-dimensional. Also, the relief is probably partially accounted for in the calculation of the radial load distribution itself. There are then a number of conceptual difficulties in making the aspect ratio analogy.

An additional uncertainty introduced by the synthesis is that it results in vary favorable ratios of lift to drag at $\alpha = 45$ degrees. This occurs in the synthesis because of the match to conditions at $\alpha = 90$ degrees, leaving the region above stall under constrained. At the 45 degree point exprimental data shows that the l/d is almost universally equal to unity, in agreement with flat plate theory.

CMA STALL MODEL

The procedure described now is an attempt to provide a more general model, one which is in agreement with stalled flat plate theory at higher angles where the flow is fully separated, which systematically generates the characteristics in the stall transition region, and over which a greater degree of parametric control can be exercised for refinement. The angle of attack range is considered in four regions as shown in Figure 17. Region 1 is the range up to stall in which the airfoil characteristics are assumed known.

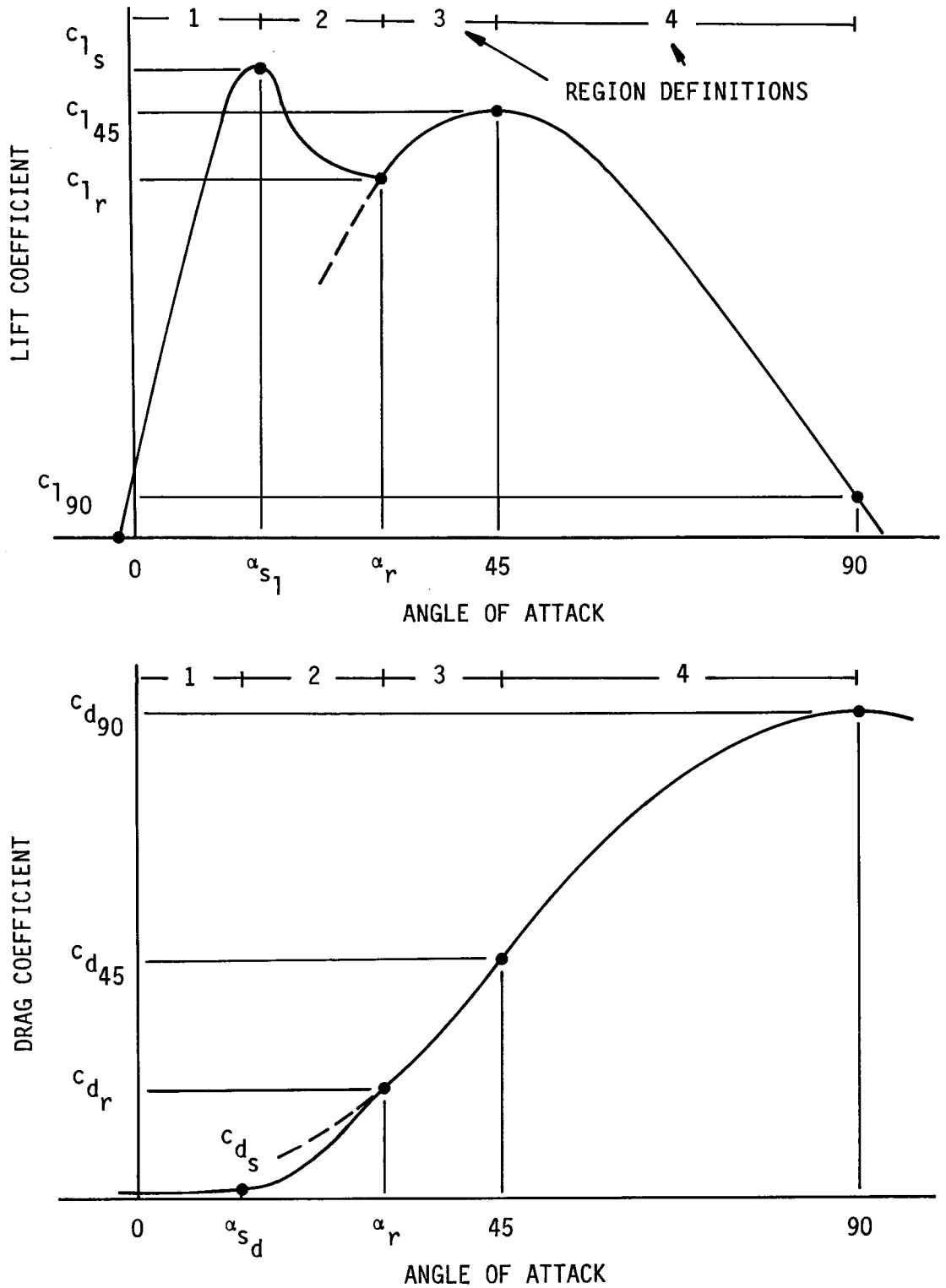


Figure 17. CMA Stall Model Region Definitions and Key Points for Matching Data.

Region 2 covers the range from stall to the recovery point after which the stalled flat plate theory is assumed to be valid. The characteristics at $\alpha = 45$ degrees are specified as parameters, as is the angle of attack for recovery. With the subscript r denoting the recovery point, then the following functions, similar to those used by Viterna, provide a very systematic fairing of the region,

$$c_{l_r} = 2c_{l_{45}}\sin(\alpha_r)\cos(\alpha_r)$$

$$c_{d_r} = 2c_{d_{45}}\sin^2(\alpha_r)$$

$$A_{12} = [c_{l_r}\cos^2(\alpha_{s_l})\sin(\alpha_r) - c_{l_s}\cos^2(\alpha_r)\sin(\alpha_{s_l})] / \{2\cos(\alpha_{s_l})\cos(\alpha_r)[\sin^2(\alpha_r)\cos(\alpha_{s_l}) - \sin^2(\alpha_{s_l})\cos(\alpha_r)]\}$$

$$A_{22} = [c_{l_s} - 2A_{12}\sin(\alpha_{s_l})\cos(\alpha_{s_l})]\sin(\alpha_{s_l})/\cos^2(\alpha_{s_l})$$

$$B_{12} = [c_{d_s}\cos(\alpha_r) - c_{d_r}\cos(\alpha_{s_d})] / [\sin^3(\alpha_{s_d})\cos(\alpha_r) - \sin^3(\alpha_r)\cos(\alpha_{s_d})$$

$$B_{22} = [c_{d_r} - B_{12}\sin^3(\alpha_r)]/\cos(\alpha_r)$$

$$c_l = 2A_{12}\sin(\alpha)\cos(\alpha) + A_{22}\cos^2(\alpha)/\sin(\alpha)$$

$$c_d = B_{12}\sin^3(\alpha) + B_{22}\cos(\alpha) \qquad (64)$$

The recovery point concept was chosen based on review of airfoil characteristics from several sources including References 16, and 17. The latter reference is particularly insightful because it indicates a fairly general trend for the family of lift curves representing Reynolds number variation to merge toward a common point at an angle of attack of about 32 degrees. This Reynolds number independence at the recovery point supports the assumption that the flow is fully separated, and that stalled flat

plate behavior represents the airfoil, independent of the pre-stall characteristics.

Region 3 then continues to 45 degrees angle of attack. The factor B_{pr} is introduced to allow determination of the effect of relieving the base (leeward side) pressure of the stalled plate. The generating functions in this region are

$$A_{13} = 2c_{l_{45}}B_{pr}$$

$$A_{23} = [c_{l_r} - 2c_{l_{45}}B_{pr}\sin(\alpha_r)\cos(\alpha_r)]/[1 - 2\sin^2(\alpha_r)]$$

$$B_{13} = [c_{d_r} - 2\,c_{d_{45}}B_{pr}\cos(\alpha_r)]/[\sin^2(\alpha_r) - (\sqrt{2}/2)\cos(\alpha_r)]$$

$$B_{23} = (c_{d_{45}}B_{pr} - B_{13}/2)\sqrt{2}$$

$$c_l = A_{13}\sin(\alpha)\cos(\alpha) + A_{23}[1-2\sin^2(\alpha)]$$

$$c_d = B_{13}\sin^2(\alpha) + B_{23}\cos(\alpha) \qquad (65)$$

And finally Region 4, with selectable values at 90 degrees angle of attack is represented by

$$c_l = B_{pr}\{2c_{l_{45}}\sin(\alpha)\cos(\alpha) - c_{l_{90}}[1-2\sin^2(\alpha)]\}$$

$$c_d = B_{pr}[c_{d_{90}}\sin^2(\alpha) + (2c_{d_{45}} - c_{d_{90}})/\sqrt{2}\cos(\alpha)] \qquad (66)$$

This model provides the capability to adjust its characteristics to match whatever is known about the actual airfoil. Or, if little information is available, then it will provide a systematic basis to synthesize the high angle of attack behavior.

EXTENSIONS FOR QUASI-STEADY SOLUTION

The analysis as described thus far is based on axial symmetry of the onset wind and the problem geometry when viewed in a fixed coordinate system attached to a reference blade. The extensions described now are designed to permit investigation at an exploratory level of effects which cause the flowfield to be time variant. The quasi-steady approach calculates the aerodynamics of the reference blade as the rotor is indexed around the azimuth, and accounts for blade motion, nonuniform wind profile, and distortion of the wake. To simplify the analysis, the effects of all other blades are calculated with the assumption that their influence is represented through the average loading over the disk.

Rigid blade flapping and cyclic pitch motions with respect to the shaft axis are permitted as harmonic functions of azimuth. The shaft may be inclined both horizontally and vertically to the principal wind direction. Nonuniformity of the onset wind is included with a ground boundary layer profile and velocity deficit functions to represent the effects of tower shadow. The wind profile as a function of height above ground y and hub height h is given by,

$$V_w = V_{w_0}[(h-y)/h]^e \tag{67}$$

Tower shadow in the form of a velocity deficit fraction T_s further modifies the wind profile to give the final axial velocity

$$V_a = V_w(1-T_s) \tag{68}$$

The velocity deficit fraction can be supplied at an array of points in the x,y- plane at the rotor disk, or the formulation described in Reference 18, can be selected.

All of these effects enter the problem through the no normal flow through the blades boundary condition, and through perturbations in the wake transport. To position a particular wake trailer, for instance, the induced component of the wake rate parameter is based on the disk average thrust coefficient developed as the quasi-steady solution stabilizes. The transport due to the distorted onset wind must now be calculated locally in the field, and therefore the position of a particular point on the trailer is found by integrating the transport from the point's origin in the field as a function of time to its position later in time (azimuth).

The working equation of the circulation solution relates the normalwash required at the blade to the induced velocity produced by all blades and wakes, and was given by Equation (11). At panel 1 on blade n Equation (11) can be expanded to isolate the contributions from each blade and wake,

$$(q_{N_1})_n = \sum_b \left[\sum_k (D_{sw})_{1,k} K_k \right]_b \tag{69}$$

With the assumption that at the reference blade all other blade's effects are through disk average loading, then the K_k circulations at all blades other than the reference blade n are taken as disk average values. By defining the circulation ratio

$$\tilde{K}_k = K_k/(K_k)_n \tag{70}$$

then Equation (69) becomes

$$(q_{N_1})_n = \sum_b \left[\sum_k (D_{sw})_{1,k} \tilde{K}_k (K_k)_n \right]_b \tag{71}$$

The influence coefficients from all the blades and wakes can now be combined as

$$(\tilde{D}_{sw})_{1,k} = \sum_b \left[(D_{sw})_{1,k} \tilde{K}_k \right]_b \tag{72}$$

giving the new working equation for the quasi-steady solution at the reference blade n as

$$(q_{N_1})_n = \sum_k (\tilde{D}_{sw})_{1,k} (K_k)_n \tag{73}$$

Since the ratio $\tilde{K}_k$ is itself a function of the solution, then an iteration is set up around the linear system solution of Equation (73), beginning with an initial value of $\tilde{K}_k = 1$. The ratio is updated after each solution of Equation (73) until the process converges. The overall solution process begins at an initial azimuth and proceeds for a specified number of steps. As the solution progresses, disk averages of the aerodynamic and performance quantities of interest are developed and can be monitored to assess the dissipation of transients as the solution approaches the (periodic) steady state.

COMPARISON TO PERFORMANCE MEASUREMENTS

The initial assessment of the lifting surface prescribed wake analysis has been made based on comparisons to measured wind turbine power characteristics and to similar calculations made with a blade element momentum theory analysis. The four turbines chosen for this comparison represent a variety of power output capacities, physical sizes, and aerodynamic configurations, and include both research and commercial machines. The turbines and major features of their configurations are listed in Table 1.

Table 1.

Turbine	Radius (ft)	Number of Blades	Airfoil Family	Chord	Twist
NASA MOD-0	63.96	2	NACA 230xx	Tapered	Untwisted
ESI-54	27.0	2	NACA 230xx	Tapered	10° Linear
Carter 25	16.0	2	NACA 230xx	Tapered	Nonlinear
Enertech 1500	6.56	3	NACA 44xx	Tapered Tip	5° Linear

Appendix D contains Program LSWT input data file listings for each of the example turbines. These listings describe the details of each turbine's blade geometry, the source and values of the airfoil performance characteristics utilized, and the selection of analysis options. As yet no attempt has been made to optimize the selected analysis parameters. Those that appear in the input listings are recommended as a nominal configuration and are consistent across all cases presented. The measured rotor power data for the NASA MOD-0, ESI-54, and Carter 25 turbines was furnished by the contracting agency. For the latter two machines the source of the data was manufacturer's tests. The measured performance of the Enertech 1500 was obtained from Reference 19. All cases are for the turbines with

blades set to zero pitch at the tip; however, information is not available to confirm that chordline reference is consistently used.

The blade element momentum analysis used for comparison is a highly modified version of the PROP Code described in Reference 6. Technical modifications include the addition of the blade element drag coefficient in the expressions used to calculate both the inplane and axial inductions. There is some disagreement over the correctness of including this term in the induction. The argument against its inclusion is based on the assumption that the drag is substantially of viscous origin, and therefore does not contribute to the lift forces. At very low lift coefficients the drag of most airfoil sections is approximately 50% viscous and 50% pressure. At the high element angles of attack at which wind turbine blades operate, the drag is mostly due to pressure. In fact, when the flow is fully separated both the lift and drag are simply components of the normal pressure force. Therefore the drag should be included in the induction.

However, the major reason for modifying the PROP code is to minimize modeling differences between it and the lifting surface analysis. To accomplish this the PROP code was restructured to accept the identical geometry input and options, to utilize the same solution grid, to accept the same airfoil data tables, and to implement the same post-stall models as the LSWT code. In most cases identical code is now used in these areas in both programs. In the comparisons that follow the modified blade element code is referred to as BEWT (Blade Element Wind Turbine).

In all cases of BEWT calculations the Prandtl tip loss option was included. Some investigators are of the opinion that any tip loss factor should be supressed if a post stall synthesis such as the Viterna model is in effect. As previously discussed this model includes an aspect ratio correction to account for three-dimensionality, and thus it can be argued that the tip loss is not required. However, that concept is strictly valid only in the context of the actuator disk theory which treats the integral loading. The blade element theory addresses radial load distribution, and thus for consistency any corrections should be conceptualized and applied

at the element level. Also, recognize that in wake methods such as the present LSWT analysis, when local blade stations are stalled the radial loading must still tend to zero at the tip. Therefore, it is again consistent to include the tip loss in the blade element analysis.

Two sets of comparisons are presented in Figures 18 through 25. The first four figures show measured rotor power as a function of wind speed for the example turbines in comparison to calculations made with the LSWT and BEWT codes, and the Viterna stall model. In the second set of four figures the calculations use the CMA stall model.

Figure 18, for the MOD-0 turbine confirms the Viterna stall model objective of achieving constant torque at high wind speeds for this turbine, based on the BEWT result. The LSWT result substantially over predicts the performance from the wind speed for peak power and higher. Also, the offset in wind speed at a given power between the LSWT and BEWT calculations at the lower wind speeds is a persistent difference for all the cases studied. Because the MOD-0 is untwisted and uses a very thick airfoil section for structural reasons, the blades stall uniformly and thus offer a striking example of the sensitivity of calculations to modeling of the high angle airfoil characteristics. It should be noted that the airfoil characteristics for this case are taken at the 0.75 radius Reynolds number, whereas the other cases utilized interpolation on Reynolds number within the airfoil tables. The increasing twist of the other example turbines results in a gentler stall propagation with wind speed. The Carter 25 calculations are in closest agreement to each other, and also to the measured data. In Figure 21, for the Enertech 1500 the LSWT calculation with the Viterna stall model appears to follow the data fairly well at the high wind speed, but the actual data at the high end was taken with as much as 30 degrees of yaw. The measurements are suspect even though they were corrected for this presentation by the cube of the cosine rule. Also, for this case the LSWT calculations at low wind speeds exhibit considerable waviness, far in excess to any appearing for the other turbines. The source of this has not been identified, but is probably an analytical limit cycle introduced by the several strong sources of nonlinearity in the problem.

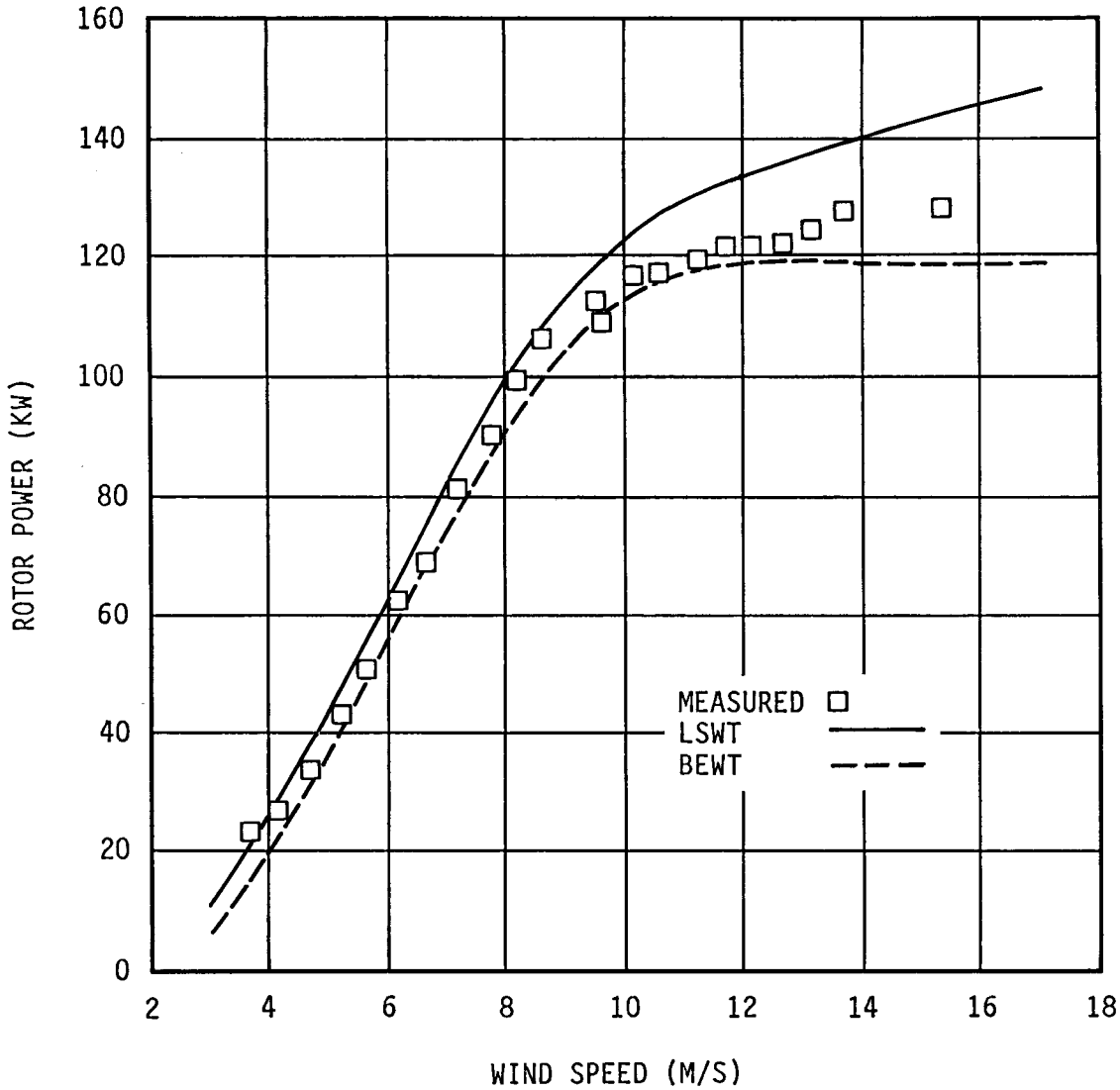


Figure 18. Comparison of Calculated and Measured Rotor Power, NASA MOD-0 Wind Turbine, Viterna Stall Model.

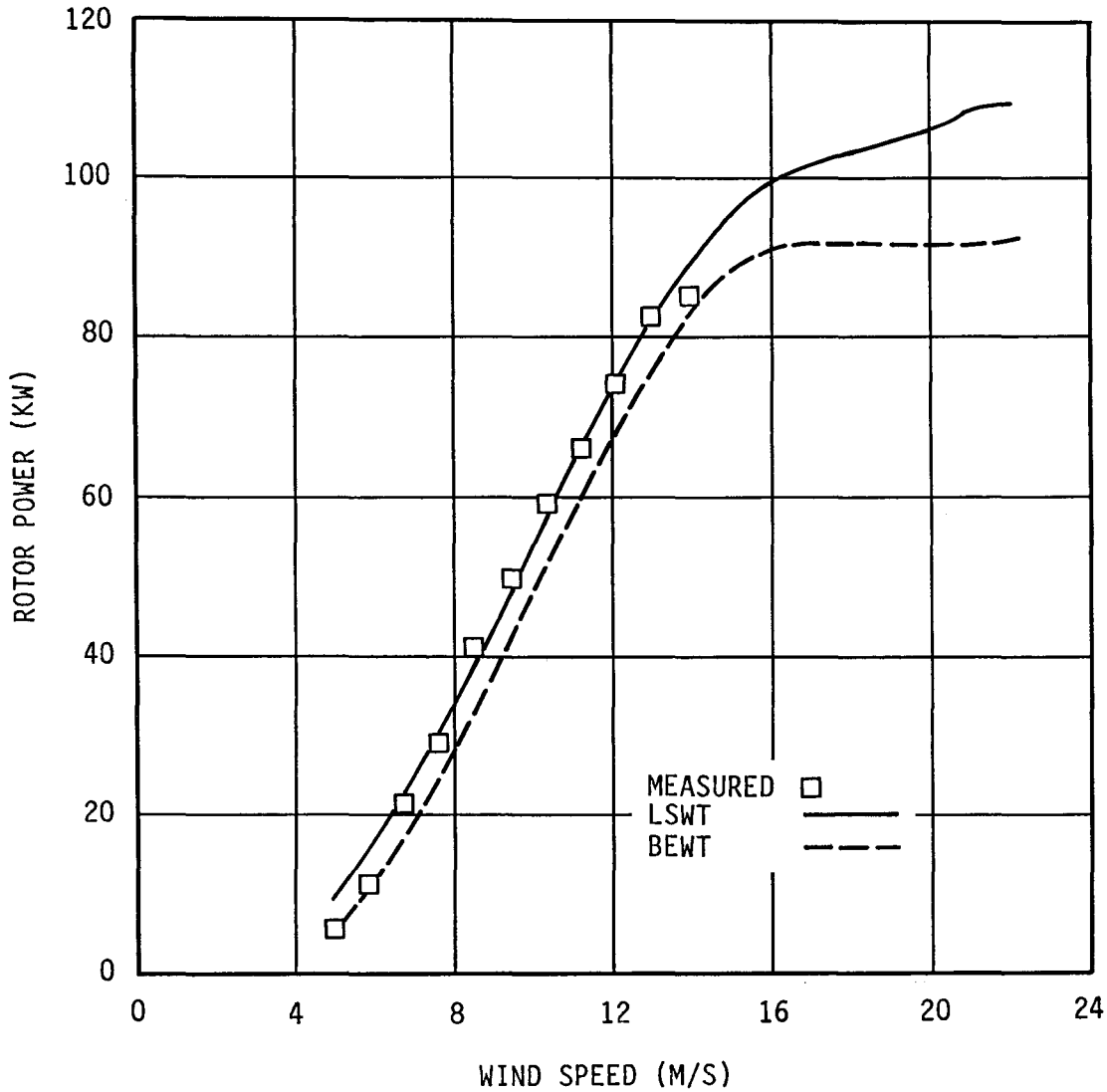


Figure 19. Comparison of Calculated and Measured Rotor Power, ESI-54 Wind Turbine, Viterna Stall Model.

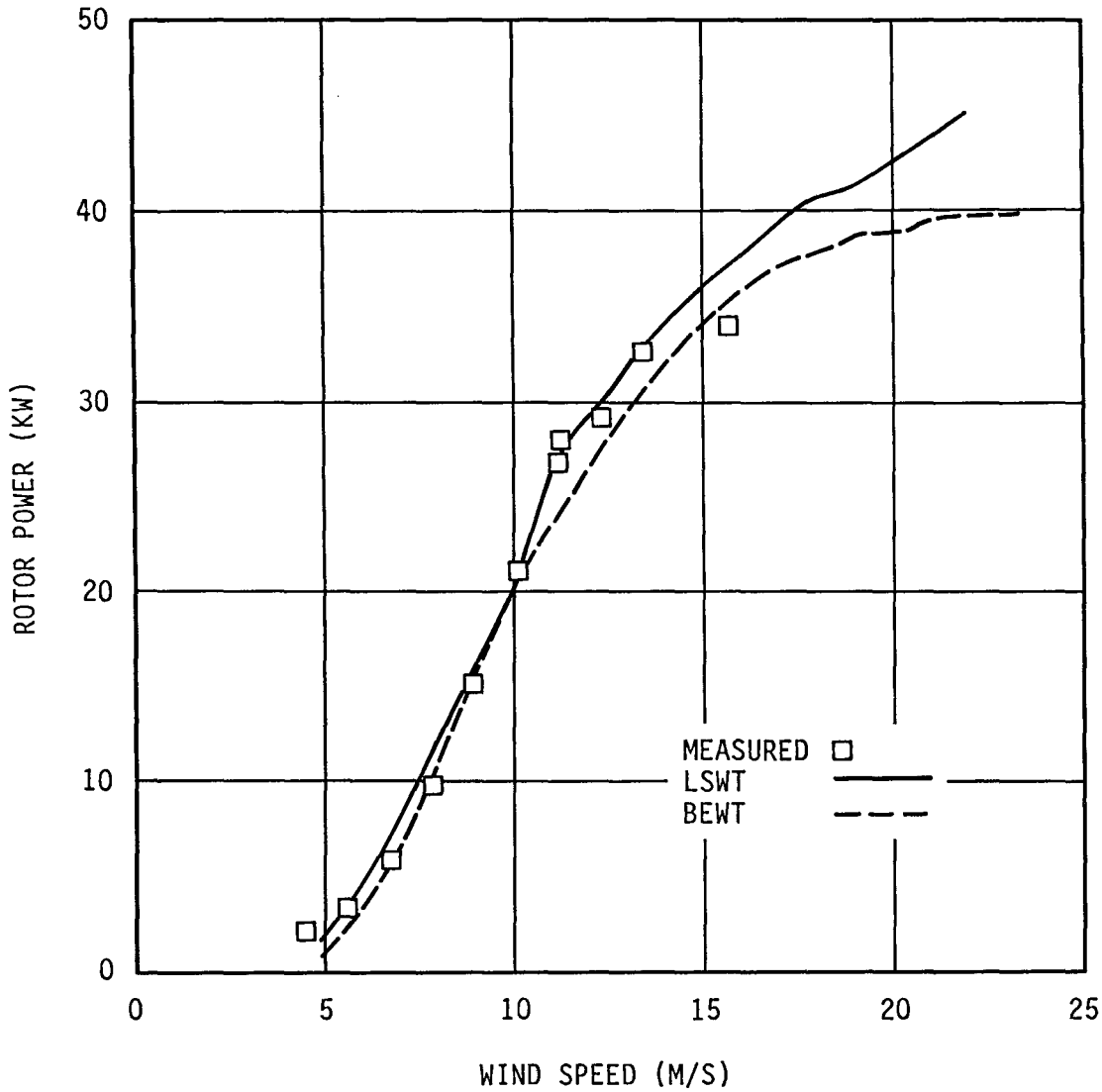


Figure 20. Comparison of Calculated and Measured Rotor Power, Carter 25 Wind Turbine, Viterna Stall Model.

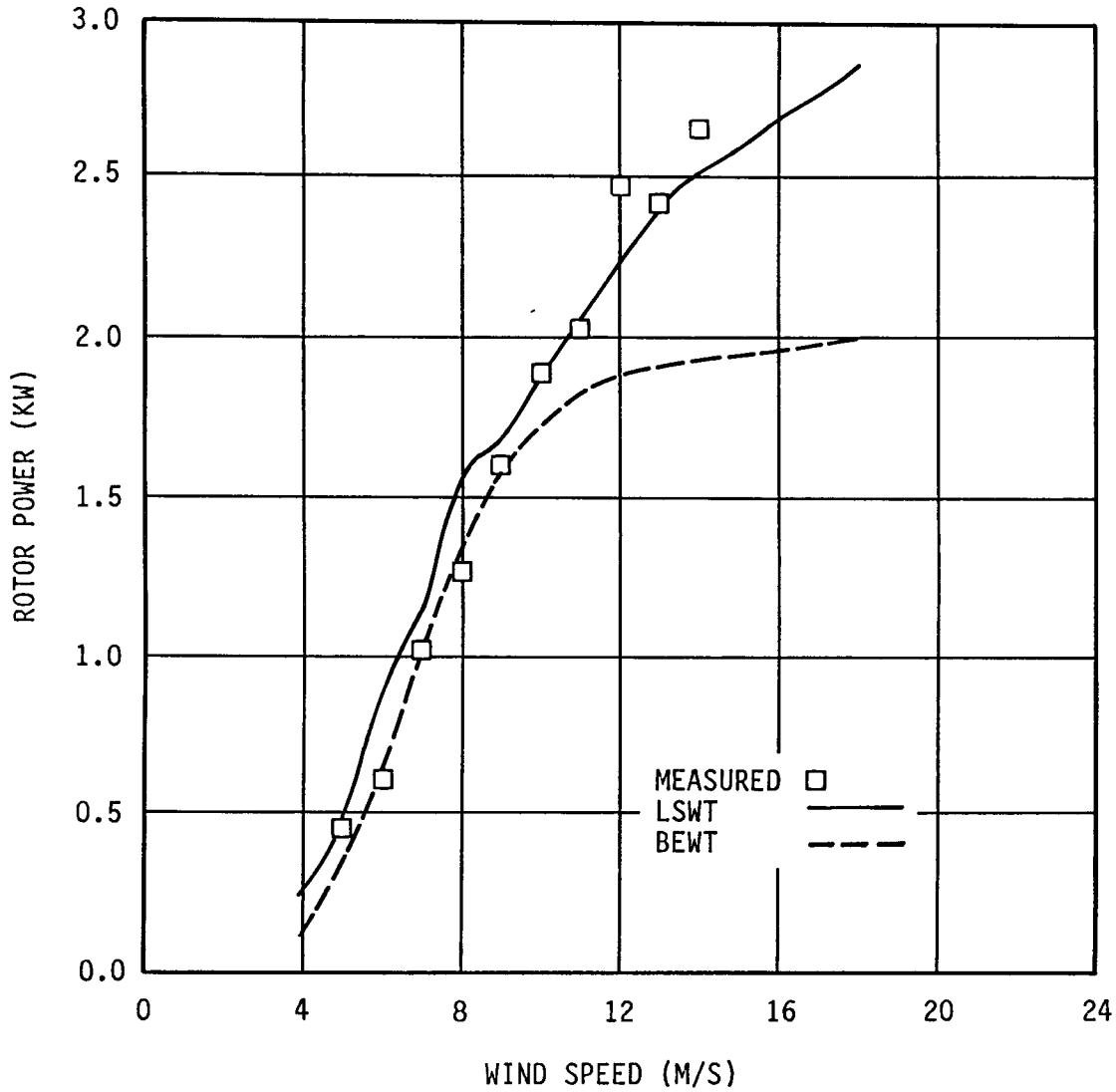


Figure 21. Comparison of Calculated and Measured Rotor Power, Enertech 1500 Wind Turbine, Viterna Stall Model.

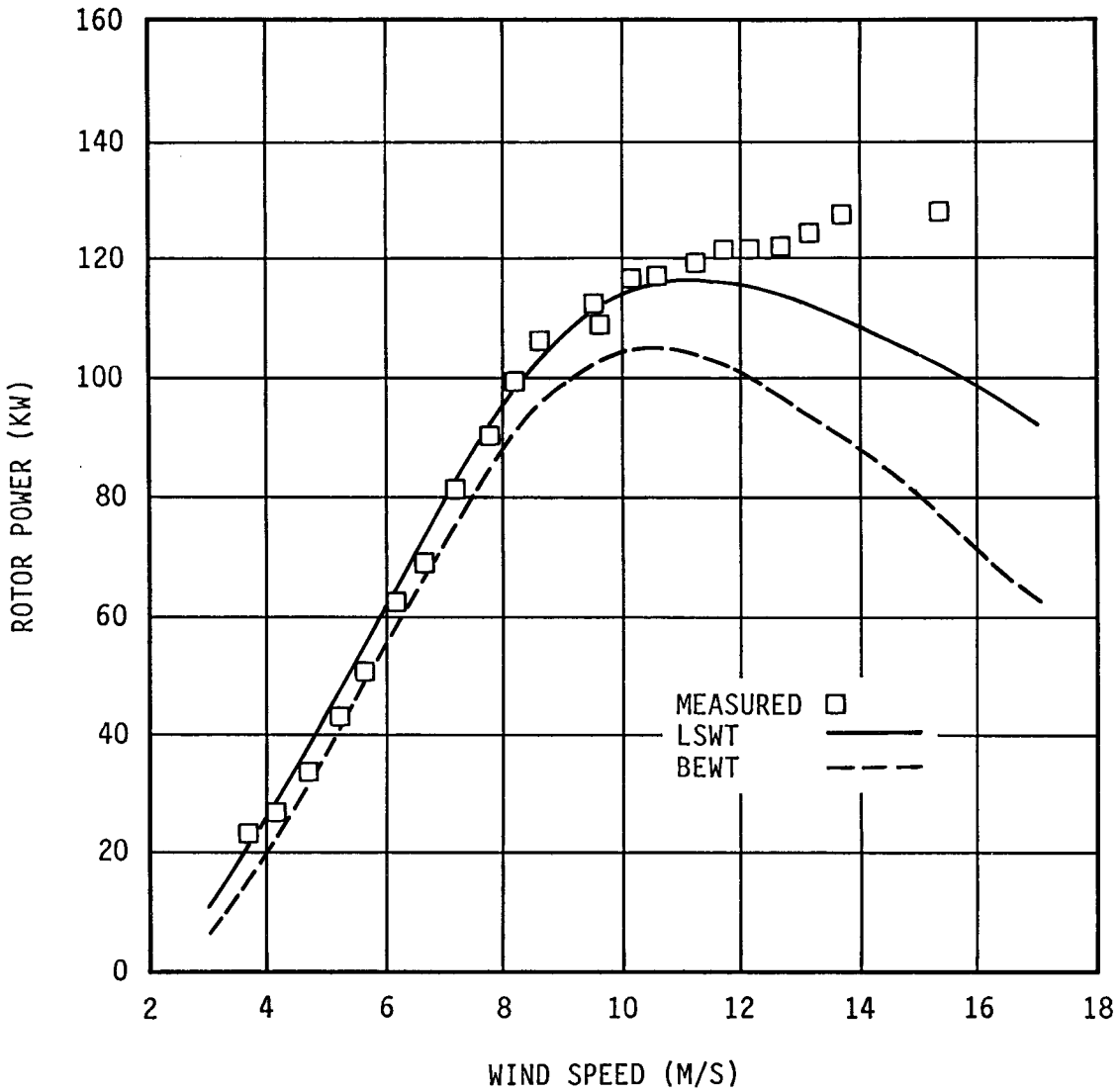


Figure 22. Comparison of Calculated and Measured Rotor Power, NASA MOD-0 Wind Turbine, CMA Stall Model.

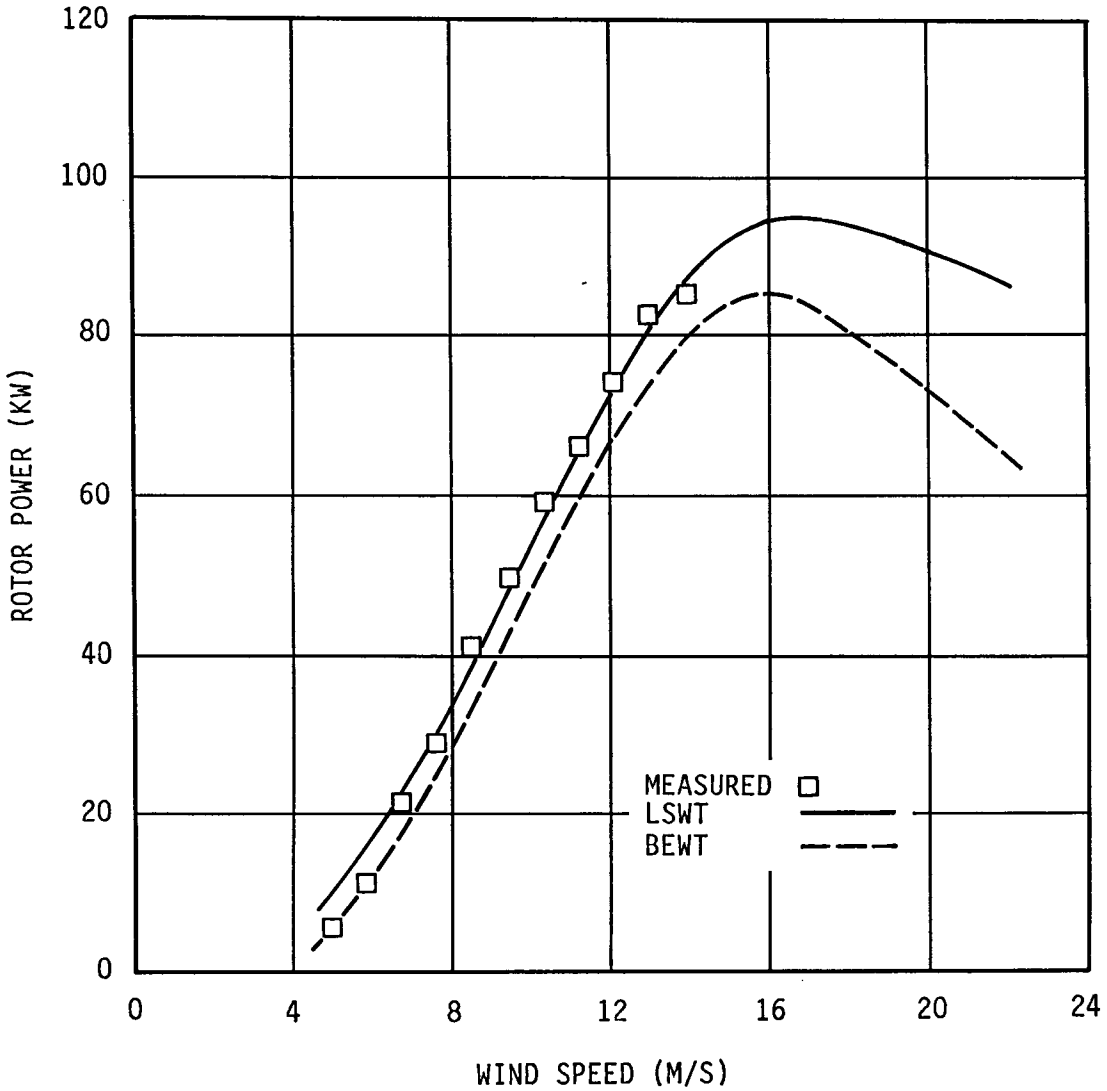


Figure 23. Comparison of Calculated and Measured Rotor Power, ESI-54 Wind Turbine, CMA Stall Model.

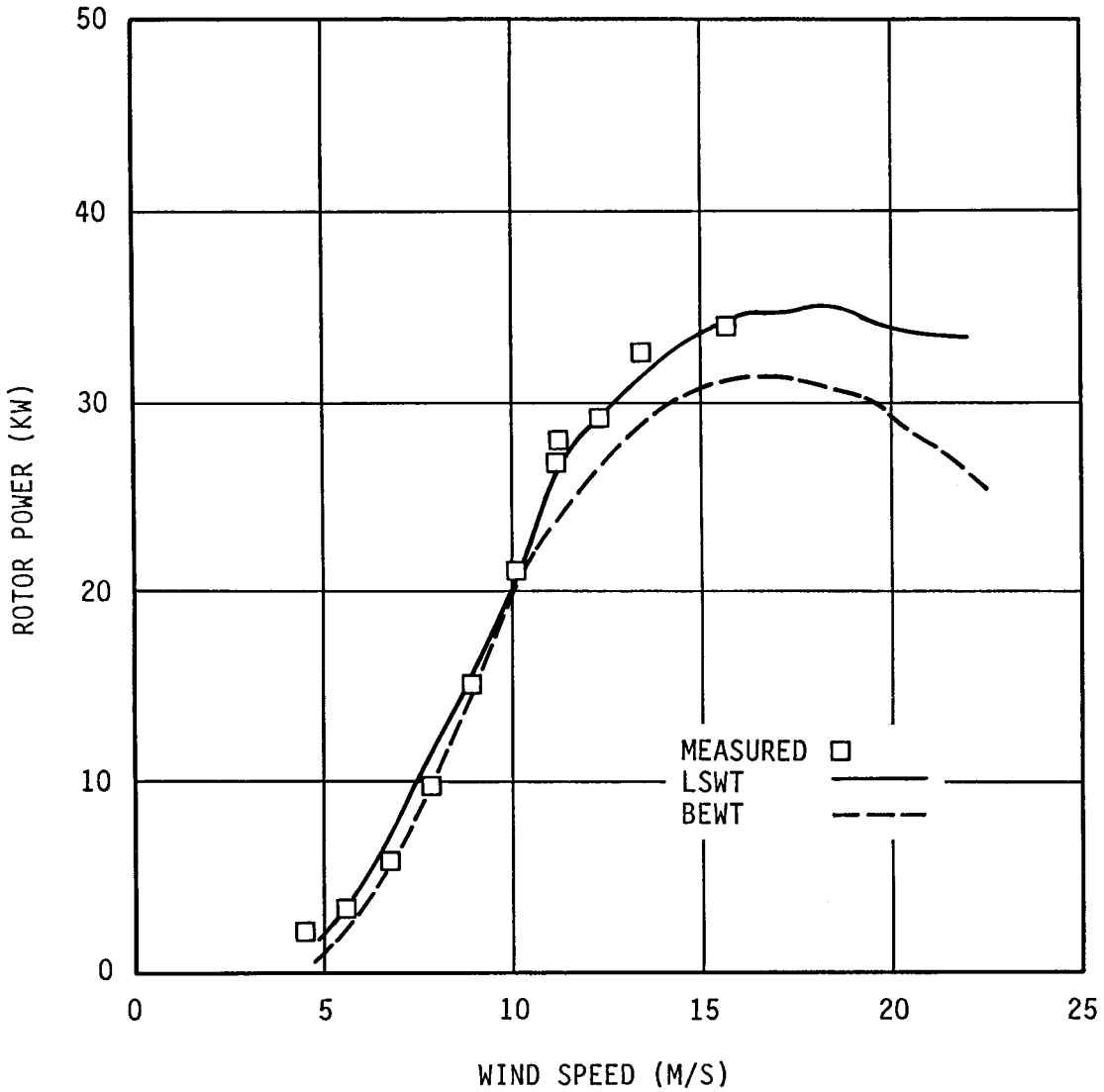


Figure 24. Comparison of Calculated and Measured Rotor Power, Carter 25 Wind Turbine, CMA Stall Model.

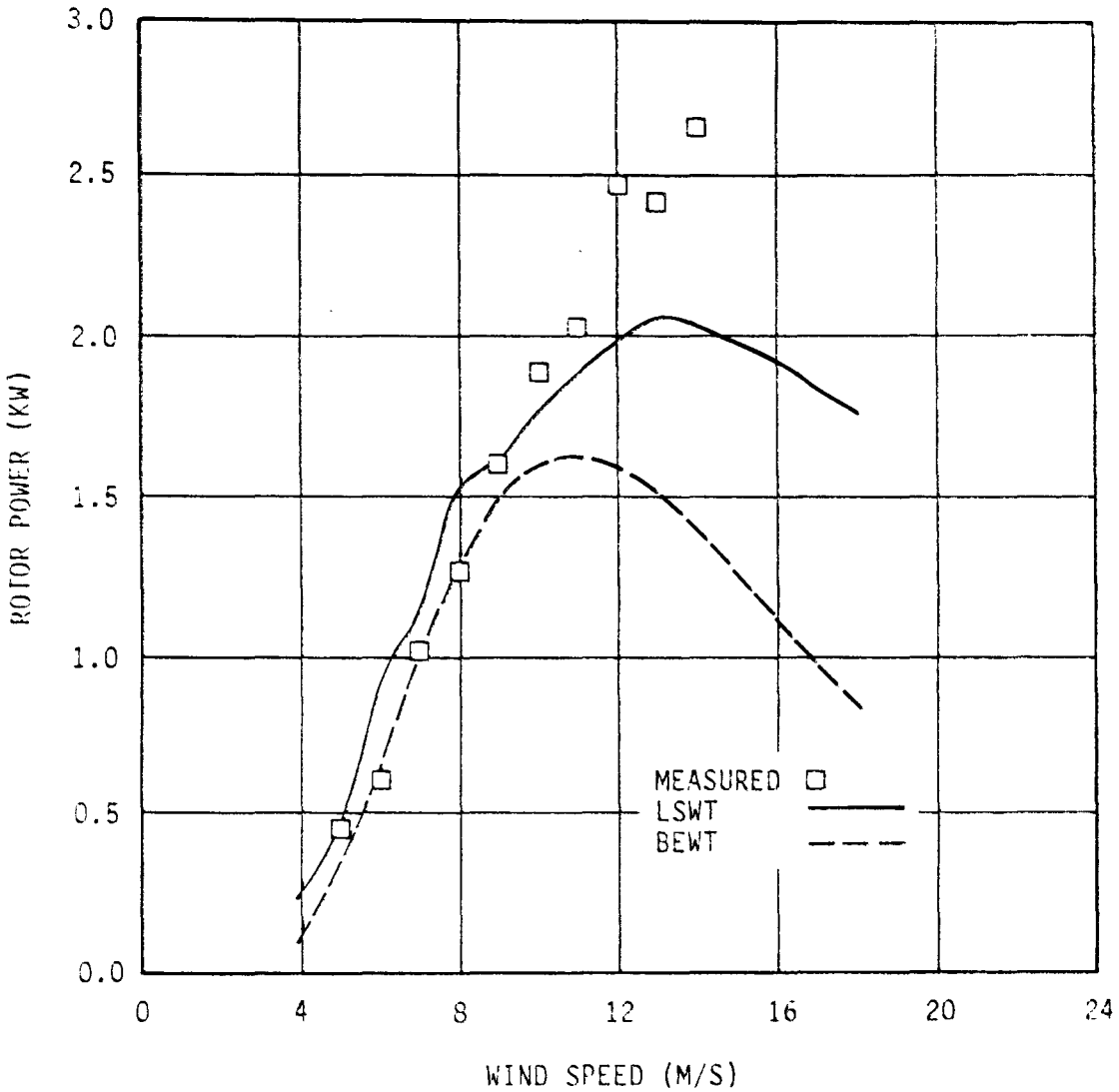


Figure 25. Comparison of Calculated and Measured Rotor Power, Enertech 1500 Wind Turbine, CMA Stall Model.

Overall the question remains as to the proper post-stall behavior and the different behavior of the two analytical methods which is emphasized when stall occurs. Since the Viterna empirical stall model was developed for use with blade element momentum methods, it is perhaps biased in that direction. Furthermore, the optimistic post stall power level calculated with the LSWT analysis is consistent with the favorable lift to drag ratios that the stall model produces in comparison to that expected from typical airfoil characteristics. This observation was in fact the motivation for developing the CMA stall model.

Figures 22 through 25, show the same comparisons as already made, but now with the new stall model. The key values in the stall model were set to realistic magnitudes guided by the flat plate characteristics given in Reference 15. This is as might be done without any knowledge of a particular airfoil's high angle behavior, resulting in

$$c_{l_{45}} = c_{d_{45}} = 1$$

$$c_{l_{90}} = 0$$

$$c_{d_{90}} = 2$$

$$\alpha_r = 32$$

In addition a value of $B_{pr} = 0.5960$, was chosen. This corresponds to the ratio of the maximum drag (at 90 degrees angle of attack) of a low aspect ratio flat plate to a two-dimensional flat plate. The reasoning is that rather than an aspect ratio effect venting the leeward side of the stalled region of a blade toward ambient pressure, the naturally occurring radial pressure gradient of the rotating system and large scale mixing would cause complete venting, independent of the position or extent of the stalled region. However, results thus far have indicated that this factor has insignificant effect for the present cases. The stall occurring in the peak power range of wind speeds is dominated by the stall transition characteristics rather than the characteristics at higher angles.

Figure 22, is the result for the MOD-0 turbine and shows that the LSWT calculation is now in much better agreement with the measured data up through peak power. At the very high wind speeds the calculations are now increasingly conservative. Similar differences with respect to the results with the Viterna model are seen for the other example cases. The overall agreement between measurements and the LSWT calculations using the new stall model for all of the example turbines is considered very good, with the exception of the high yaw angle data of the Enertech 1500 as previously noted. This suggests that the ability to predict peak power with the present method is improved over the blade element momentum method. Although these calculations should be viewed as preliminary, it is emphasized that careful modeling of the physical configurations in the LSWT analysis combined with straight forward estimates of the airfoil behavior produced these results.

CONCLUSIONS AND RECOMMENDATIONS

A comprehensive lifting surface analysis for horizontal axis wind turbine rotors has been developed and implemented, and the initial validation conducted for the prediction of steady rotor power output. The method uses a prescribed wake model extended from a helicopter hovering wake model with modifications for the axial flow regime of the wind turbine. The analysis now provides the opportunity to investigate and define the sensitivity of wind turbine aerodynamic calculations and performance estimates to much finer details of modeling than was previously possible. The important effects of proper representation of airfoil characteristics in the stall transition region was demonstrated and a new stall simulation model was developed from which refinements can be systematically made.

Several recommendations are in order to pursue further development of the method, and to clarify a number of physical issues, especially in regard to understanding and modeling stall aerodynamics. First, validation of the lifting surface method should be broadened and include detailed evaluation of modeling options, particularly those associated with the wake itself. With the flexibility of the present wake model it should be possible to establish the minimum model required to produce reliable results with minimum demand on computational resource.

Second, stall aerodynamics should be investigated to understand the theoretical requirements to extend both momentum theory and potential flow analysis methods into the stall regime. In conjunction with this, a better understanding of airfoil stall transition behavior should be sought, most probably guided by experimental investigations, the goal of which would be to establish a baseline from which methods to account for stall progression and three-dimensionality on the blade can be developed.

The importance of vortex ring work state induced velocity should be clarified, and the significant differences between the Glauert empirical model and conflicting experimental data should be reconciled.

Finally, to support validation of methods such as developed in the present work, a wind turbine data base should be established which contains documented physical characteristics, airfoil aerodynamic coefficients with sufficient definition in Reynolds's number and stall behavior, and substantiated benchmark aerodynamic and performance measurements for several representative wind turbines.

REFERENCES

1. Kocurek, J.D., Berkowitz, L.F., and Harris, F.D., "Hover Performance Methodology at Bell Helicopter Textron," Presented at the 36th Annual National Forum of the American Helicopter Society, Washington, D.C., Paper No. 80-3-1, May 1980.

2. Kocurek, J.D., A Lifting Surface Performance Analysis with Circulation Coupled Wake for Advanced Configuration Hovering Rotors, Ph.D. Dissertation, Texas A&M University, May 1978.

3. Kocurek, J.D., Tangler, J.L., " A Prescribed Wake Lifting Surface Hover Performance Analysis," Presented at the 32nd Annual National Forum of the American Helicopter Society, Washington, D.C., Paper No. 1001, May 1976.

4. Glauert, H., "Airplane Propellers," Vol. IV, Div. L of Aerodynamic Theory, W.F. Durand, ed., Julius Springer, Berlin, 1935.

5. Wilson, R.E., Lissaman, P.B.S., and Walker, S.M., "Aerodynamic Performance of Wind Turbines," Oregon State University, Research Applied to National Needs (RANN), under Grant no. AER74-04014 A03, 1976.

6. Hibbs, B., and Radkey, R.L., Calculating Rotor Performance with the Revised Prop Code, AeroVironment, Inc., Pasadena, California, Contract No. PFN-13470W, Prepared for Rockwell International Corporation, Energy Systems Group, Rocky Flats Plant, Wind Energy Research Center, Golden, Colorado, March 1984.

7. Castles, Jr.,W., and Gray, R.B., Empirical Relation Between Induced Velocity, Thrust, and Rate of Descent of a Helicopter Rotor as Determined by Wind-Tunnel Tests on Four Model Rotors, NACA TN 2474, Washington, D.C., October 1951.

8. Gray, R.B., An Aerodynamic Analysis of a Single Bladed Rotor in Hovering and Low Speed Forward Flight as Determined from the Smoke Studies of the Vorticity Distribution in the Wake, Princeton Aeronautical Engineering Department Report No. 356, September 1956.

9. Landgrebe, A.J., An Analytical and Experimental Investigation of Helicopter Rotor Hover Performance and Wake Geometry Characteristics, USAAMRDL Technical Report 71-24, Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia, June 1971.

10. Egolf, T.A. and Landgrebe, A.J., "The UTRC Wind Energy Conversion System Performance Analysis for Horizontal Axis Wind Turbines (WECSPER)," Presented at the 2nd DOE/NASA Wind Turbine Dynamics Workshop, Cleveland, Ohio, February 1981.

11. Miller, R.H., Simplified Free Wake Analysis for Rotors, ASRL TR 194-3, Aeroelastic and Structures Research Laboratory, Department Aeronautics and Astronautics, Massachusetts, August 1981.

12. Glauert, H., "The Analysis of Experimental Results in the Windmill Brake and Vortex Ring States of an Airscrew," R. and M. No. 1026, British A.R.C.,February 1926.

13. Drees, J. M., Lucassen, L. R., and Hendal, W. P., "Airflow Through Helicopter Rotors in Vertical Flight," N.L.L. Report V.1535, December 1955.

14. Viterna, L. A. and Corrigan, R. D., "Fixed Pitch Rotot Performance of Large Horizontal Axis Wind Turbines," presented at the DOE/NASA Workshop on Large Horizontal Axis Wind Turbines, Cleveland, Ohio, July, 1981.

15. Hoerner, S. F., Fluid Dynamic Drag, Published by the Author, 1965.

16. Abbott, I.H., and von Doenhoff,A.E., Theory of Wing Sections, Dover Publications, Inc., 1958,pp. 8-30.

17. Jacobs, E. N. and Sherman, A., Airfoil Section Characteristics as Affected by Variations of the Reynolds Number, NACA Report 586.

18. Thresher, R.W., and Hershberg, E.L., Computer Analysis Of Wind Turbine Blade Static And Dynamic Loads, Mechanical Engineering Department, Oregon State University, Contract No. ASC76824JR, Prepared for Rockwell International Corporation, Energy Systems Group, Rocky Flats Plant, Wind Energy Research Center, Golden, Colorado, March 1984.

19. Tangler, J. L., Horizontal-Axis Wind System Rotor Performance Model Comparison, A Compendium, RFP-3508 UC-60, Rockwell International Corporation, February, 1983.

APPENDIX A

LSWT INPUT SEQUENCE GUIDE

LSWT INPUT SEQUENCE GUIDE

The input data set for computer program LSWT is organized into four groups of records described as INPUT SETS 1 through 4. INPUT SET 1 contains case set-up, control, and option parameters. The second set describes blade geometry in terms of piecewise linear radial distributions of planform parameters such as chord and pitch increment due to twist. The distribution of airfoils is also contained in this set.

The blade configuration of the turbine is described by as many as four independent segments. Each segment requires the input sequence in INPUT SET 2. Segment boundaries should be chosen to correspond to physical details of the blade such as changes in chord taper ratio, twist rate, and airfoil designation. Because the blade need not be continuous, the segmentation scheme can also be used to model actual breaks in the geometry as might correspond to a step change in local pitch.

INPUT SET 3 is optional and provides the additional information and options to run LSWT in quasi-steady mode. The fourth set contains bivarient tables of two-dimensional airfoil characteristics as functions angle of attack and Reynold's number, along with parameters for the several analytical stall models which may be selected.

In the following sections, record-by-record descriptions of the input set to LSWT are described. Each record contains a maximum of 72 characters, thus allowing for sequence numbers to be placed in columns 73-80. All numeric data, including integer switch and option values, is input in floating point notation in fields of ten columns formated as F10.0, with a maximum of seven fields per record.

INPUT SET 1, CASE SET-UP AND CONTROL OPTIONS

RECORD 1 (72A1) Case Identification

FIELD	NAME	DESCRIPTION
1-72	CASID	user notation to be displayed on report pages

RECORD 2A (7F10.0) Print Options Switches

1-10	IPRT1	surface grid and collocation point coordinates
	0.0	option off
	1.0	print each azimuth
	2.0	print initial azimuth only
	3.0	print reference blade position only
11-20	IPRT2	wake coordinates
	0.0	option off
	1.0	print each cycle
	2.0	print each azimuth
	3.0	print final azimuth only
21-30	IPRT3	blade element performance summary
	0.0	option off
	1.0	print each cycle
	2.0	print final cycle only
31-40	IPRT4	blade performance summary
	0.0	option off
	1.0	print each cycle
	2.0	print final cycle only
41-50	IPRT5	rotor performance summary
	0.0	option off
	1.0	print each cycle
	2.0	print final cycle only

51-60 IPRT6 wake parameter summary

0.0	option off
1.0	print each cycle
2.0	print final cycle only

61-70 IPRT7 controls printing of program trace markers to show timing and flow through major modules and subroutines.

0.0	option off
>0.0	print program trace to unit pointed to by IPRT7

RECORD 2B (7F10.0) Print Options, continued

1-10 IPRT8 controls printing of record image of input data set to the output listing

0.0	option off
1.0	print image of input data set

RECORD 3 (7F10.0) Tolerances

Program default tolerances are selected by entering a zero value for any tolerance. Any non-zero entry will be used to override the default. Convergence of any parameter occurs when the absolute value of the quantity

$$\frac{\text{(new value) - (old value)}}{\text{(new value)}} \leq \text{tolerance}$$

1-10 TOL1 convergence tolerance of Gauss-Seidel iterative solution for circulation. The default is 1.0E-05 for the inner solution. The outer solution tolerance is (TOL1)*10.

11-20 TOL2 sectional effective linear lift curve slope tolerance. The default is a function of the lift slope and is given by

1.0/ALS**2

or set to 1.0E-02, whichever is greater.

21-30 TOL3 wake parameter tolerance (axial settling rate). The default is 5.0E-02

RECORD 4 (7F10.0) Iteration Cycle Limits

Program default iteration cycle limits are selected by entering a zero value for any limit. Any non-zero entry will be used to override the default. The particular iteration will be halted after the specified number of cycles, regardless of the convergence state.

1-10	ICYL1	Gauss-Seidel iterative solution for circulation. The default for he inner solution is 100 cycles. The limit for the outer solution is (ICYL1)/5.
11-20	ICYL2	sectional effective linear lift curve slope. The default is four cycles.
21-30	ICYL3	wake parameters. The default is six cycles.
31-40	ICYL4	tip vortex rollup. The default is four cycles.

RECORD 5 (7F10.0)

1-10	ISTDY	Steady/Quasi-Steady Mode Switch
	0.0	case is steady, INPUT SET 3 must be omitted
	1.0	case is quasi-steady, INPUT SET 3 is required
11-20	RPM	rotor rotational speed (revolutions per min)
21-30	AMURAT	ratio of atmospheric viscosity to the standard day, sea-level value, $\mu_0 = 3.719 \times 10^{-7}$ slugs/ft-sec.
31-40	SIGMAP	ratio of atmospheric density to the standard day, sea-level value, $\rho_0 = 0.002378$ slugs/ft^3

RECORD 6 (7F10.0) Wind Speed Sweep Options

1-10	VWOIN	initial value of wind speed (dimension corresponds to value of VDIM input below)
11-20	DVWO	sweep increment of wind speed (dimension corresponds to value of VDIM input below)
21-30	NVWO	number of points in sweep
31-40	VDIM	Wind speed dimension switch
	0.0	ft/sec
	1.0	meters/sec
	2.0	miles per hour
	3.0	tip speed ratio

RECORD 7 (7F10.0) Rotor Collective Pitch

1-10	ALPSIN	initial value of blade pitch (deg)
11-20	DALP	sweep increment of blade pitch (deg)
21-30	NALP	number of points in sweep
31-40	THRUST	initial value of thrust (lbs)

RECORD 8 (7F10.0) Rotor Parameters

1-10	NBLDS	number of blades
11-20	MP	number of chordwise panels per blade ($1 \leq MP \leq 5$)
21-30	NSGMNT	number of blade segments per blade ($1 \leq NSGMNT \leq 4$)
31-40	RADIUS	reference radius used to make geometry and parameters non-dimensional (ft)

RECORD 9 (7F10.0) Wake Options

1-10	REVS	number of wake revolutions in discrete wake region
11-20	NWIPR	number of wake increments per revolution (this value will be automatically adjusted in the first revolution to provide a fine grid). Note: $(REVS + 1) * NWIPR + 1 \leq 133$
21-30	IFWAK	far wake option
	0.0	analytical far wake influence <u>is not</u> included in induced velocity calculations
	1.0	analytical far wake influence <u>is</u> included in induced velocity calculations.
31-40	PSIMR	azimuth increment over which inboard sheet will develop into root vortex
41-50	PSIMT	azimuth increment over which outboard sheet will develop into tip vortex Assure that values are chosen for PSIMR and PSIMT that clear the blade's trailing edge, and that complete vortex development within 360 deg.
51-60	IROLL	roll-up iteration option. Active only if outboard sheet and either mid- or inboard sheets are selected in RECORD 10.
	0.0	iteration supressed
	1.0	compatibility of tip vortex circulation and number of rolled-up wake filaments will be assured through iteration (ICYL4)

RECORD 10 Wake Sheet Filament Indices For Roll-up Specification

The wake may be divided into three sheet regions, an inboard sheet, a mid-sheet, and an outboard sheet. The bounding filaments for each sheet region are specified by their indices in RECORD 10. Any sheet region may be omitted by setting its bounding indices to zero. If an inboard or outboard sheet is selected, it will develop into a rolled-up vortex over the azimuth increment PSIMR or PSIMT, respectively, as given in RECORD 9. At least one sheet must be selected. Bounding indices must be sequential

At least one sheet must be selected. Bounding indices must be sequential from root to tip, and must be exclusive to a single region.

1-10	JS(1,1)	index of inboard filament of inboard sheet
11-20	JS(1,2)	index of outboard filament of inboard sheet
21-30	JS(2,1)	index of inboard filament of mid-sheet
31-40	JS(2,2)	index of outboard filament of mid-sheet
41-50	JS(3,1)	index of inboard filament of outboard sheet
51-60	JS(3,2)	index of outboard filament of outboard sheet

INPUT SET 2, BLADE SEGMENT DESCRIPTIONS

The blade is described by as many as four independent segments. Each segment requires the input sequence in INPUT SET 2. NSGMNT segment descriptions are required. The segments are numbered from inboard to outboard, and the segment descriptions must also be input in inboard to outboard sequence.

RECORD 1 (7F10.0) Segment Options And Parameters

1-10	NPS	number of spanwise panels on segment (The maximum number of panels for the blade is twenty. The minimum for each segment is one.)
11-20	NGRID	spanwise grid (panel boundary) option for segment
	1.0	equal spacing
	2.0	full cosine distribution
	3.0	half cosine, density increased inboard
	4.0	half cosine, density increased outboard
21-30	XOCREF	non-dimensional chord reference of input quantities
31-40	YSEG(1)	inboard extent of segment, (ft)
41-50	YSEG(2)	outboard extent of segment, (ft)

RECORD 2 (7F10.0) Segment Orientations

1-10	XTRAN	translations (ft) to place segment in desired position
11-20	YTRAN	
21-30	ZTRAN	
31-40	XROT	rotations (deg) to place segment in desired position
41-50	YROT	
51-60	ZROT	

RECORD 3 (7F10.0) Segment Chord Distribution

1-10 NCHORD number of chord inputs ($2 < NCHORD < 24$)

RECORD 4A [, 4B [, 4C [, 4D]]] (7F10.0)

1-10 YCHORD(1) blade stations (ft) at which piecewise linear chord distribution is specified

11-20 YCHORD(2)

... as required

61-70 YCHORD(7)

Continue with RECORDS 4B, 4C, AND 4D as required to input NCHORD values of YCHORD.

RECORD 5A [, 5B [, 5C [, 5D]]] (7F10.0)

1-10 CHORD(1) value of blade chord (ft) corresponding to the YCHORD stations input in RECORD 4

11-20 CHORD(2)

... as required

61-70 CHORD(7)

Continue with RECORDS 5B, 5C, and 5D as required to input NCHORD values of CHORD.

RECORD 6 (7F10.0) Segment Twist Distribution

1-10 NTWIST number of twist inputs ($2 \leq NTWIST \leq 24$)

RECORD 7A [, 7B [, 7C [, 7D]]] (7F10.0)

1-10	YTWIST(1)	blade stations (ft) at which piecewise linear twist distribution is specified
11-20	YTWIST(2)	
...	as required	
61-70	YTWIST(7)	

Continue with RECORDS 7B, 7C, AND 7D as required to input NTWIST values of YTWIST.

RECORD 8A [, 8B [, 8C [, 8D]]] (7F10.0)

1-10	TWIST(1)	value of blade twist increment (deg) corresponding to the YTWIST stations input in RECORD 7
11-20	TWIST(2)	
...	as required	
61-70	TWIST(7)	

Continue with RECORDS 8B, 8C, and 8D as required to input NTWIST values of TWIST.

RECORD 9 (7F10.0) Segment Planform Offset Distribution

1-10	NXAOFF	number of planform offset inputs ($2 \leq$ NXAOFF ≤ 24)

RECORD 10A [, 10B [, 10C [, 10D]]] (7F10.0)

1-10	YXAOFF(1)	blade stations (ft) at which piecewise linear planform sweep offset distribution is specified
11-20	YXAOFF(2)	

... as required

61-70 YXAOFF(7)

Continue with RECORDS 10B, 10C, AND 10D as required to input NXAOFF values of YXAOFF.

RECORD 11A [, 11B [, 11C [, 11D]]] (7F10.0)

1-10 XAOFF(1) values of blade planform offset (ft) corresponding to the YXAOFF stations input in RECORD 10

11-20 XAOFF(2)

... as required

61-70 XAOFF(7)

Continue with RECORDS 11B, 11C, AND 11D as required to input NXAOFF values of XAOFF.

RECORD 12 (7F10.0) Segment Airfoil Distribution

1-10 NAFT number of airfoil region definition inputs, ($1 \leq NAFT \leq 24$)

RECORD 13A, ... (7F10.0)

1-10 IDAF(1) three digit airfoil table identification code

11-20 YAFT(1,1) inboard extent of this airfoil (ft)

21-30 YAFT(1,2) outboard extent of this airfoil (ft)

Continue with RECORDS 13B, ... as required to input NAFT records.

INPUT SET 3, Quasi-Steady Mode

Input set 3 is only required if ISTDY = 1 (Input set 1, Record 5), otherwise it must be omitted.

RECORD 1 (7F10.0) Azimuth Sweep

1-10	PSIRIN	initial value of azimuth at which blade 1 will be placed (deg)
11-20	DPSIR	sweep increment (deg)
21-30	NPSIR	number of points in sweep

RECORD 2 (7F10.0) Shaft Orientation and Wind Profile

1-10	ALPS1	shaft inclination with respect to wind, positive shaft down into wind.
11-20	ALPS2	shaft yaw with respect to wind, positive shaft right into wind
21-30	HO	hub height (ft) above ground
31-40	E	exponent of shape function for atmospheric boundary layer
41-50	ITSTYP	tower shadow model option
	0.0	tower shadow not included
	1.0	tower shadow equations used
	2.0	tower shadow table used

RECORD 3A (7F10.0) Tower Shadow Velocity Deficit Fraction Equations. Input Required If ITSTYP = 1, otherwise omit this RECORD.

For the equation option, the tower shadow velocity deficit fraction is defined from

$$T_s(\psi) = t_0 + t_p \cos[p(\psi - \psi_s)] \quad \text{for } \psi_s - \psi_0 < \psi < \psi_s + \psi_0$$

		$T_s(\psi) = 0$, elsewhere
1-10	TO	t_0
11-20	TP	t_p
21-30	TPO	ψ_0 (deg)
31-40	TPS	ψ_s (deg)
41-50	TK	k, defines $p = 2\pi k/(2\psi_0)$

RECORD 3B (7F10.0) Tower Shadow Velocity Deficit Fraction Table. Input Required If ITSTYP = 2, otherwise omit RECORDS 3B-5B.

1-10	NXTS	number of X-grid entries
11-20	NYTS	number of Y-grid entries

The table size is based on the combination of NXTS and NYTS subject to the constraint

$$8 \leq NXTS+NYTS+(NXTS*NYTS) \leq 100$$

The program stores the table internally in the array TSTAB(I) where table members and indices are as follows:

X-grid locations	$1 \leq I \leq NXTS$
Y-grid locations	$(NXTS+1) \leq I \leq (NXTS+NYTS)$
velocity deficit coefficients	$(NXTS+NYTS+1) \leq I \leq (NXTS+NYTS+(NXTS*NYTS))$

The minimum table requires four grid point locations (two each of X-grid and Y- grid) and four velocity deficit coefficients. The program will interpolate the table for a velocity deficit coefficient for all blade collocation point locations and for all wake filament trajectory points. If the program requires a value of the velocity deficit coefficient outside the rectangle specified by the grid locations, then it will be assumed that there is no deficit for that point.

RECORD 4A, ... (10X,6F10.0) Tower Shadow Table X-Grid Locations

11-20	TSTAB(1)	values of X-grid locations (ft)
21-30	TSTAB(2)	
...	as required	
61-70	TSTAB(7)	

Continue with RECORDS 4B, ... as required to input NXTS values of the X-grid location

RECORD 5A (7F10.0) Tower Shadow Table Y-Grid Locations

1-10	TSTAB(NXTS+1)	value of Y-grid location (ft)
11-20	TSTAB(NXTS+NYTS+1)	velocity deficit coefficients
21-30	TSTAB(NXTS+NYTS+2)	
...	as required	
61-70	TSTAB(NXTS+NYTS+6)	

Continue with RECORDS 5B, ... as required to input NXTS values of velocity deficit coefficients for this Y-grid location.

RECORD 5B, ... (10X,6F10.0) Velocity Deficit Fraction Coefficients, Continued

11-20	TSTAB(NXTS+NYTS+7)	velocity deficit fraction coefficients
21-30	TSTAB(NXTS+NYTS+8)	
...	as required	
61-70	TSTAB(NXTS+NYTS+12)	

Continue with RECORDS 5C, ... as required to input NXTS values of velocity deficit coefficients for this Y-grid location.

RECORD 6 (7F10.0) Flapping Motion Coefficients

1-10	NBH	number of harmonics in series ($0 \leq NBH \leq 5$)
11-20	AB(1,1)	constant coefficient

RECORD 7 (7F10.0) Sine Terms

1-10	AB(2,1)	coefficients of sine terms (deg)
11-20	AB(3,1)	
...	as required to input NBH coefficients	
41-50	AB(6,1)	

RECORD 8 (7F10.0) Cosine Terms

1-10	AB(2,2)	coefficients of cosine terms (deg)
11-20	AB(3,2)	
...	as required to input NBH coefficients	
41-50	AB(6,2)	

RECORD 9 (7F10.0) Cyclic Motion Coefficients

1-10	NTH	number of harmonics in series ($0 \leq NTH \leq 5$)
11-20	AT(1,1)	constant coefficient

RECORD 10 (7F10.0) Sine Terms

1-10	AT(2,1)	coefficients of sine terms (deg)
11-20	AT(3,1)	
... as required to input NTH coefficients		
41-50	AT(6,1)	

RECORD 11 (7F10.0) Cosine Terms

1-10	AT(2,2)	coefficients of cosine terms (deg)
11-20	AT(3,2)	
... as required to input NTH coefficients		
41-50	AT(6,2)	

INPUT SET 4, AIRFOIL DATA TABLES

Two-dimensional bivarient airfoil tables provide lift, drag, and pitching moment coefficient characteristics as functions of angle of attack and Reynold's number. Each airfoil data table requires the input sequence in INPUT SET 4. The number of tables required depends upon the airfoil descriptions specified in INPUT SET(S) 2. At least one table must be input. The program is dimensioned to accomodate a maximum of four tables. Each table's size is such that any combination of its size parameters (RECORD 3 of this set) must meet the constraints

$$NACL+NRCL+(NACL*NRCL) \leq 200$$

$$NACD+NRCD+(NACD*NRCD) \leq 200$$

$$NACM+NRCM+(NACM*NRCM) \leq 100$$

The program stores each table internally in a column (JAFT) of the arrays CxTAB(I,JAFT), where 'x' represents 'L' for a lift table , 'D' for a drag table or 'M' for a pitching moment. The storage array column contains the table members in the following sequence:

Reynolds numbers	$1 \leq I \leq NRCx$
Angles Of Attack	$(NRCx+1) \leq I \leq (NRCx+NACx)$
Aero-Coefficients	$(NRCx+NACx+1) \leq I \leq (NRCx+NACx+(NRCx*NACx))$

The minimum size table requires NACx = 2, and NRCx = 1. In this case the interpolation on Reynold's number is suppressed and the one value of Reynold's number input is for reference only. Also the table must contain suficient data so that the angle for zero lift may be determined.

RECORD 1 (7F10.0) Number Of Tables, Stall Option

1-10	NAFT	number of tables to be input ($1 \leq NAFT \leq 4$)
11-20	IAFTEX	analytical stall region option
	0.0	tables will be extrapolated for angles outside of range
	1.0	stalled flat plate model
	2.0	Viterna's stall model
	3.0	CMA stall model

RECORD 2 (F10.0, A62) Airfoil Table ID

1-10	IDAFT(1)	table identifier (3-digit number)
11-72	TITLE	descriptive information

RECORD 3 (7F10.0) Table Size Parameters

1-10	NRCL	number of Reynolds numbers in lift table
11-20	NACL	number of angles of attack in lift table
21-30	NRCD	number of Reynolds numbers in drag table
31-40	NACD	number of angles of attack in drag table
41-50	NRCM	number of Reynolds numbers in moment table
51-60	NACM	number of angles of attack in moment table

RECORD 4, ... (7F10.0) Stall Model Parameters

This record is included to allow convenient access to stall model parameters used in the present implementation, and for future refinement of these models. The first field in the record is the thickness to chord ratio for the airfoil described by the present table. Six subsequent fields are available to pass parameters to the stall models. The current input requirements are dependent upon the selected stall model as shown.

1-10 AFTOC thickness to chord ratio of airfoil

For IAFTEX = 0.0 or 1.0, no further input is required for this record.

For IAFTEX = 2.0, input the blade aspect ratio to be used in the Viterna stall model. No further input is required for this record.

11-20 AR

For IAFTEX = 3.0, input the stall region delimiters used in the CMA stall model.

11-20	ALPR	angle of attack (deg) at which table coefficients recover to and follow the analytical model
21-30	CL45	lift coefficient at 45 degrees
31-40	CD45	drag coefficient at 45 degrees
41-50	CL90	lift coefficient at 90 degrees
51-60	CD90	drag coefficient at 90 degrees
61-70	BPR	base pressure relief factor which will multiply coefficients at angles above ALPR

RECORD 5A, ... (10X,6F10) Lift Table Reynolds Numbers

11-20	CLTAB(1)	values of Reynolds number, ratioed to 10^6
21-30	CLTAB(2)	
...	as required	
61-70	CLTAB(6)	

Continue with RECORDS 5B, ... as required to input NRCL values of Reynolds number.

RECORD 6A (7F10) Angle Of Attack And Lift Coefficients

1-10	CLTAB(NRCL+1)	angle of attack (deg)
11-20	CLTAB(NRCL+NACL+1)	lift coefficients for this angle of attack
21-30	CLTAB(NRCL+NACL+2)	
...	as required	
61-70	CLTAB(NRCL+NACL+6)	

Continue with RECORDS 6B, ... as required to input NRCL values of lift coefficient for this angle of attack.

RECORD 6B (10X,6F10) Lift Coefficients, Continued

11-20	CLTAB(NRCL+NACL+7)	lift coefficients for this angle of attack, continued
21-30	CLTAB(NRCL+NACL+8)	
...	as required	
61-70	CLTAB(NRCL+NACL+12)	

Continue with RECORDS 6C, ... as required to input NRCL values of lift coefficient for this angle of attack.

Repeat RECORDS 6 format for next angle of attack and corresponding lift coefficients, until NACL angles of attack have been input.

RECORD 7A, ... (10X,6F10) Drag Table Reynolds Numbers

11-20	CDTAB(1)	values of Reynolds number, ratioed to 10^6
21-30	CDTAB(2)	
...	as required	
61-70	CDTAB(6)	

Continue with RECORDS 7B, ... as required to input NRCD values of Reynolds number.

RECORD 8A (7F10) Angle Of Attack And Drag Coefficients

1-10	CDTAB(NRCD+1)	angle of attack (deg)
11-20	CDTAB(NRCD+NACD+1)	drag coefficients for this angle of attack
21-30	CDTAB(NRCD+NACD+2)	

... as required

61-70 CDTAB(NRCD+NACD+6)

Continue with RECORDS 8B, ... as required to input NRCD values of drag coefficient for this angle of attack.

RECORD 8B (10X,6F10) Drag Coefficients, Continued

11-20	CDTAB(NRCD+NACD+7)	drag coefficients for this angle of attack, continued
21-30	CDTAB(NRCD+NACD+8)	
... as required		
61-70	CDTAB(NRCD+NACD+12)	

Continue with RECORDS 8C, ... as required to input NRCD values of drag coefficient for this angle of attack.

Repeat RECORDS 8 format for next angle of attack and corresponding drag coefficients, until NACD angles of attack have been input.

RECORD 9A, ... (10X,6F10) Moment Table Reynolds Numbers

11-20	CMTAB(1)	values of Reynolds number, ratioed to 10^6
21-30	CMTAB(2)	
... as required		
61-70	CMTAB(6)	

Continue with RECORDS 9B, ... as required to input NRCM values of Reynolds number.

RECORD 10A (7F10) Angle of Attack And Moment Coefficients

1-10	CMTAB(NRCM+1)	angle of attack (deg)
11-20	CMTAB(NRCM+NACM+1)	moment coefficients for this angle of attack

21-30 CMTAB(NRCM+NACM+2)

... as required

61-70 CMTAB(NRCM+NACM+6)

Continue with RECORDS 10B, ... as required to input NRCM values of moment coefficient for this angle of attack.

RECORD 10B (10X,6F10) Moment Coefficients, Continued

11-20 CMTAB(NRCM+NACM+7) moment coefficients for this angle of attack, continued

21-30 CMTAB(NRCM+NACM+8)

... as required

61-70 CMTAB(NRCM+NACM+12)

Continue with RECORDS 10C, ... as required to input NRCM values of moment coefficient for this angle of attack.

Repeat RECORDS 10 format for next angle of attack and corresponding moment coefficients, until NACM angles of attack have been input.

APPENDIX B

PROGRAM AND INSTALLATION NOTES

PROGRAM AND INSTALLATION NOTES

The Program LSWT source code is written in FORTRAN and should be installable on most host systems with only minor modifications for compatibility with system dependent interfaces. Program LSWT was developed and is currently implemented under the Microsoft FORTRAN compiler, Version 3.2, for the MS-DOS Operating System on the IBM PC class of microcomputers. Language extensions have been avoided so that the program can be installed using any FORTRAN compiler designed to the '66 or '77 standard. Program LSWT is computationally intensive and therefore fully expanded hardware including a floating point accelerator is highly recommended.

The LSWT source code is maintained in eight files or modules of functionally related subprogram units. Table B-1 describes each module and lists its subprograms. This grouping arrangement and file sizing are recommended for convenient program maintenance and compatibility with editors. A complete listing of Program LSWT is provided as Appendix C.

SYSTEM INTERFACE

Source module LSWT07 contains all code required to interface Program LSWT to the host system. This module is presently designed toward a foreground execution environment as typically found on a single-user microcomputer system. All files are managed from within the program, and particular file specifications for standard input and output are prompted from and supplied by the user at execution time. To accomplish this, subroutines CLRSCN and LOCATE are provided for console screen handling. these use standard escape sequences for ANSI compatible terminal devices. Note that the use of this capability on an IBM PC type computer requires the ANSI.SYS device driver be installed when the operating system is booted. For other terminal types the screen handling routines must be modified appropriately. For Program LSWT installation in a batch type environment these interfaces may be completely eliminated, provided that the program

TABLE B-1. LSWT PROGRAM MODULE SUMMARY

Module/Subroutine	Function
LSWT01.FOR	Executives and Utilities
MAIN	Main Program and Executive
INTLS	Initialize Lift Curve Slopes
EXINT	Initialize Program
EXBLD	Generate Reference Blade Geometry
EXAIC	Aerodynamic Influence Coefficients
EXSLV	Lifting Surface Solution
BVTINT	Bi-Varient Table Interpolation
ZROLFT	Determine Zero Lift LIne From Airfoil Tables
LSWT02.FOR	Lifting Surface Core Analysis
AICDLS	Calculate 2-D Aerodynamic Influence Coefficients
AICDS	Calculate Lifting Surface AIC's
AICDW	Calculate Wake AIC's
AICDSW	Assemble AIC Matrix From Components
GSITER	Solve Linear Equations By Gauss-Seidel Iteration
VLINE	Calculate Velocity Induced By Line Vortex
VCYLN	Calculate Velocity Induced By Vortex Cylinder
LSWT03.FOR	Blade And Rotor Geometry
GRID	Generate Blade Surface Grid System
COLCRD	Calculate Location Of Collocation Points
RFSURF	Develop Reference Blade Surface Geometry
AFTLOC	Determine Airfoil Table Indices At Solution Stations
LINTRP	Perform Linear Interpolation Of Geometry Parameters
ROTATE	Perform Rotation Transformation Of Coordinates
PSURF	Place Reference Blade At Target Position
VPANG	Calculates The Sine Of Angle Between Two Vectors
LSWT04.FOR	Prescribed Wake Geometry
PREWAK	Generate Prescribed Wake
FILMNT	Calculate Wake Filament Coordinates
FILROL	Calculate Rolled-Up Wake Filament Coordinates
PWAKE	Position Reference Wake Coordinates
WAKPRM	Determine Prescribed Wake Parameters
WAKUPD	Update Wake Parameters
ROLNDX	Update Wake Sheet Indices For Roll-Up Iteration

TABLE B-1 (CONTINUED)

Module/Subroutine	Function
LSWT05.FOR	Program Input
READIN	Main Input Subroutine
READAF	Set Up To Read Airfoil Tables
REDTAB	Read Tables
LSWT06.FOR	Program Output Reports
OUT1	Report Reference Blade Surface Coordinates
OUT2	Report Reference Blade's Wake Coordinates
OUT3	Report Blade Element Aerodynamic Characteristics
OUT4	Report Blade Performance Summary
OUT5	Report Rotor Performance Summary
OUT6	Report Wake Parameters And Work State
OUT8	Report Record Image Of Input File
HEADER	Write Banner And Case Information On Report Pages
APPEND	Append IOBUFFER File To Standard Output File
LSWT07.FOR	System Interface
IOINIT	Initialze I/O System
IOFNSH	Close I/O System
TRACE	Generate Program Position And Timing Markers
TIMER	Interface To System Clock
LOCATE	Position Console Display Cursor
CLRSCN	Clear Console Display
LSWT08.FOR	Aerodynamic Calculations
WNDPRO	Determine Wind Profile
FLPVEL	Calculate Blade Flapping Velocity
NRMLBC	Impose Normalwash Boundary Condition
CIRPRM	Calculate Circulation Parameters
BEVEL	Calculate Blade Element Velocities
CWAVG	Calculate Chordwise Parameter Average
BEALPH	Calculate Blade Element Angles Of Attack
AFCOEF	Calculate Airfoil Aerodynamic Coefficients
AFTEXT	Calculate Airfoil Post Stall Characteristics
AEROPF	Calculate Aerodynamic Performance

file system (FORTRAN Unit Numbers) can be associated with external devices or files on external devices, and managed from the operating system's command language.

Program timing information is available from subroutine TIMER. Within TIMER, a call is made to system dependent subroutine SYSTIM which reads the system clock and returns the time in hours, minutes, seconds, and hundredths of seconds. TIMER calculates lapsed time and reports it to the output stream. The SYSTIM interface subroutine (or a dummy routine) must be provided for the particular host system. This is the only non-FORTRAN code required by Program LSWT. An implementation of SYSTIM written in assembly language for the IBM PC family of processors is included in Appendix C.

LSWT FILE SYSTEM

The file system is initialized in subroutine IOINIT where default unit numbers are assigned to integer variable unit pointers. These variable pointers are passed to the subprograms requiring them through a common block. All program references to I/O units are made to these pointers, thus the actual unit number assignment is only required in IOINIT. The variables used for the I/O units and the function of each file are outlined in Table B-2.

IOU1 is associated with the system console device for both input and output. The remaining units fall into two categories. IOU5 through IOU8 are the usual sequential, formatted I/O files. IOU10 through IOU13 are auxiliary mass storage files required for temporary retention of large arrays during program execution. These files are direct access and unformatted. The record lengths and formulas for calculation of actual space requirements for these files is included in the commented listing of IOINIT. The use of this file structure minimizes program internal storage requirements by requiring, for instance, only one column of the influence coefficient matrix of the linear equation system to be resident at any time.

TABLE B-2. FILE SYSTEM DESCRIPTION

Unit	Use
IOU1	Console
IOU5	Standard Input
IOU6	Standard Output
IOU7	Trace File (Not An Actual File, Only Points to Another Output Unit)
IOU8	Temporary Output Buffer
IOU10	Mass Storage File For Blade Surface AIC's
IOU11	Mass Storage File For Wake AIC's
IOU12	Mass Storage File For Combined AIC Array
IOU13	Mass Storage File For Wake Coordinates

COMMON BLOCK STRUCTURE

Program LSWT common blocks are maintained in individual files and combined with the source code through INCLUDE statements at compile time. The format of the file names is the actual name of a particular block with the extension '.CMN' appended. This insures that common variables share the same name and dimensions across all program units. Also, rather than listing the included common files directly in a particular subprogram, a single include file for each subprogram is used, and this file lists the individual common files. The naming convention for these list files uses the subprogram name with the extension '.CML' appended. The common block file names and the common list file names along with their contents are given in Table B-3 and Table B-4, respectively.

TABLE B-3. COMMON BLOCK FILES

FILE: 'AFTABL.CMN'

```
      COMMON/ AFTABL / NAFT,    IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
```

FILE: 'ALSLP.CMN'

```
      COMMON/ ALSLP  / ALS(20),        ALSNEW(20),     ALSOLD(20)
```

FILE: 'BELOAD.CMN'

```
      COMMON/ BELOAD / DFX(20),        DFXI(20),       DFXP(20),
     1                 DFZ(20),        DFZI(20),       DFZP(20),
     2                 DQ(20),         DQI(20),        DQP(20),
     3                 DPM(20)
```

FILE: 'BLDELE.CMN'

```
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
```

FILE: 'BLLOAD.CMN'

```
      COMMON/ BLLOAD / CTRB,    CQRB,    CTWB,    CPWB,    BTHRST, BTORQ,
     1                 BTORQI, BTORQP, BPWR,    BPWRP,   BPWRI,  BPMOMT
```

FILE: 'CALPHA.CMN'

```
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),
     2                 TWSTCG(20)
```

FILE: 'CASE.CMN'

```
      COMMON/ CASE   / IPOINT, ALPHAD, VWOD,    IVDIM,   VWFAC,  PSIRD,
     1                 CASEID
```

FILE: 'CCIRC.CMN'

```
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
```

FILE: 'CDLS.CMN'

```
      COMMON/ CDLS   / DLS(25)
```

FILE: 'CGRID.CMN'

```
      COMMON/ CGRID  / NGRID(4),       XG(6),  YG(24), XCG(5)
```

FILE: 'CIPRT.CMN'

```
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
```

FILE: 'CONFIG.CMN'

```
      COMMON/ CONFIG / XOCREF(4),      YSEG(4,2),      JCHORD(4,2),
     1                 YCHORD(24),     CHORD(24),      JTWIST(4,2),
     2                 YTWIST(24),     TWIST(24),      JXAOFF(4,2),
     3                 YXAOFF(24),     XAOFF(24),      JAFT(4,2),
     4                 YAFT(24,2),     IDAF(24),       XTRAN(4),
     5                 YTRAN(4),       ZTRAN(4),       XROT(4),
     6                 YROT(4),        ZROT(4)
```

FILE: 'CONSTS.CMN'

```
      COMMON/ CONSTS / PI,     RPD
```

FILE: 'CONTRL.CMN'

```
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
```

FILE: 'CTCON.CMN'

```
      COMMON/ CTCON  / CTRIN, CTROUT, CTRUPD
```

TABLE B-3 (CONTINUED)

FILE: 'CTIMER.CMN'

```
      COMMON/ CTIMER / HO,     MO,     SO,     DO,     H1,     M1,
     1                 S1,     D1
```

FILE: 'CVLCTY.CMN'

```
      COMMON/ CVLCTY / VFS1(5,20),     VFS2(5,20),     VFS3(5,20),
     1                 VFS1M(20),      VFS2M(20),      VFS3M(20),
     2                 VI1(5,20),      VI2(5,20),      VI3(5,20),
     3                 VI1M(20),       VI2M(20),       VI3M(20),
     4                 VREL(20),       VIND(20),       QREL(20)
```

FILE: 'CWIND.CMN'

```
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
```

FILE: 'GVINT.CMN'

```
      COMMON/ GVINT  / XA,     YA,     ZA,     XB,     YB,     ZB,
     1                 XC,     YC,     ZC,     AN1C,   AN2C,   AN3C,
     2                 AT1C,   AT2C,   AT3C
```

FILE: 'IFLAGS.CMN'

```
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
```

FILE: 'NMWSH.CMN'

```
      COMMON/ NMWSH  / NRMWSH(100)
```

FILE: 'PARMS.CMN'

```
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
```

FILE: 'PNLCRD.CMN'

```
      COMMON/ PNLCRD / XS(6,24),       YS(6,24),       ZS(6,24),
     1                 XCOL(5,20),     YCOL(5,20),     ZCOL(5,20)
```

FILE: 'PNLNDX.CMN'

```
      COMMON/ PNLNDX / MP,            NP,            NPANEL,
     1                 MP1,           NP1,           NPNLSQ,
     2                 NSGMNT,        JGLIM(4,2),    JCGLIM(4,2),
     3                 NPS(4),        NBLDS
```

FILE: 'PNLVEC.CMN'

```
      COMMON/ PNLVEC / AN1(5,20),     AN2(5,20),     AN3(5,20),
     1                 AT1(5,20),     AT2(5,20),     AT3(5,20),
     2                 AQC1(20),      AQC2(20),      AQC3(20)
```

FILE: 'REFSUR.CMN'

```
      COMMON/ REFSUR / XSR(6,24),     YSR(6,24),     ZSR(6,24),
     1                 XCOLR(5,20),   YCOLR(5,20),   ZCOLR(5,20),
     2                 AN1R(5,20),    AN2R(5,20),    AN3R(5,20),
     3                 AT1R(5,20),    AT2R(5,20),    AT3R(5,20)
```

FILE: 'RTLOAD.CMN'

```
      COMMON/ RTLOAD / CTRR,   CQRR,   CTWR,   CPWR,   RTHRST, RTORQ,
     1                 RTORQP, RTORQI, RPWR,   RPWRP,  RPWRI,  CTR,
     2                 CQR,    CTW,    CPW,    THRST,  TORQ,   TORQP,
     3                 TORQI,  PWR,    PWRP,   PWRI,   RPMOMT, PMOMT
```

FILE: 'SWEEP.CMN'

```
      COMMON/ SWEEP  / ALPSIN, DALP,   NALP,   VWOIN,  DVWO,   NVWO,
     1                 PSIRIN, DPSIR,  NPSIR
```

FILE: 'UNITS.CMN'

```
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
```

FILE: 'WAKCRD.CMN'

```
      COMMON/ WAKCRD / XW(133,24),    YW(133,24),    ZW(133,24)
```

TABLE B-3 (CONCLUDED)

FILE: 'WAKNDX.CMN'

```
      COMMON/ WAKNDX / REVS,          NWIPR,          NWI,
     1                 KMAX(24),      IFWAK,          JS(3,2),
     2                 PSIMR,         PSIMT,          IROLL
```

FILE: 'WRATES.CMN'

```
      COMMON/ WRATES / AKZW,   AKZI,   AKZ,    AKR,    A,      AKB,
     1                 CTRW,   AKZO,   ALAMO,  AKRO,   VAXB,   VNUB
```

TABLE B-4. COMMON BLOCK LIST FILES

```
FILE: 'AEROPF.CML'

$INCLUDE: 'BELOAD.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'BLLOAD.CMN'
$INCLUDE: 'CALPHA.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CTCON.CMN'
$INCLUDE: 'CVLCTY.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'RTLOAD.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'AFCOEF.CML'

$INCLUDE: 'AFTABL.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CONSTS.CMN'


FILE: 'AFTEXT.CML'

$INCLUDE: 'AFTABL.CMN'
$INCLUDE: 'CALPHA.CMN'
$INCLUDE: 'CONSTS.CMN'


FILE: 'AFTLOC.CML'

$INCLUDE: 'AFTABL.CMN'
$INCLUDE: 'CONFIG.CMN'


FILE: 'AICDLS.CML'

$INCLUDE: 'CDLS.CMN'
$INCLUDE: 'CGRID.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'CCIRC.CMN'
```

TABLE B-4 (CONTINUED)

```
FILE: 'AICDS.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'GVINT.CMN'
$INCLUDE: 'PNLCRD.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'PNLVEC.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'AICDSW.CML'

$INCLUDE: 'ALSLP.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CDLS.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'AICDW.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'GVINT.CMN'
$INCLUDE: 'PNLCRD.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'PNLVEC.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKCRD.CMN'
$INCLUDE: 'WAKNDX.CMN'


FILE: 'BEALPH.CML'

$INCLUDE: 'ALSLP.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CALPHA.CMN'
$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CONTRL.CMN'
$INCLUDE: 'CVLCTY.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'
```

```
FILE: 'BEVEL.CML'

$INCLUDE: 'ALSLP.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CDLS.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CVLCTY.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'PNLVEC.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'CIRPRM.CML'

$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'PNLNDX.CMN'


FILE: 'CLRSCN.CML'

$INCLUDE: 'UNITS.CMN'


FILE: 'COLCRD.CML'

$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CGRID.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'CWAVG.CML'

$INCLUDE: 'CGRID.CMN'
$INCLUDE: 'PNLNDX.CMN'


FILE: 'EXAIC.CML'

$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PNLNDX.CMN'
```

TABLE B-4 (CONTINUED)

```
FILE: 'EXBLD.CML'

$INCLUDE: 'CIPRT.CMN'


FILE: 'EXINI.CML'

$INCLUDE: 'CONSTS.CMN'


FILE: 'EXINT.CML'

$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONSTS.CMN'


FILE: 'EXSLV.CML'

$INCLUDE: 'CASE.CMN'
$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONTRL.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKNDX.CMN'


FILE: 'FILMNT.CML'

$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PNLCRD.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKCRD.CMN'
$INCLUDE: 'WAKNDX.CMN'
$INCLUDE: 'WRATES.CMN'
```

```
FILE: 'FILROL.CML'

$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLCRD.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKCRD.CMN'
$INCLUDE: 'WAKNDX.CMN'


FILE: 'FLPVEL.CML'

$INCLUDE: 'CWIND.CMN'
$INCLUdE: 'PARMS.CMN'


FILE: 'GRID.CML'

$INCLUDE: 'CGRID.CMN'
$INCLUDE: 'CONFIG.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLNDX.CMN'


FILE: 'GSITER.CML'

$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CONTRL.CMN'
$INCLUDE: 'NMWSH.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'HEADER.CML'

$INCLUDE: 'CASE.CMN'


FILE: 'INTLS.CML'

$INCLUDE: 'AFTABL.CMN'
$INCLUDE: 'ALSLP.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CALPHA.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'
```

TABLE B-4 (CONTINUED)

```
FILE: 'IOFNSH.CML'

$INCLUDE: 'UNITS.CMN'


FILE: 'IOINIT.CML'

$INCLUDE: 'UNITS.CMN'


FILE: 'LOCATE.CML'

$INCLUDE: 'UNITS.CMN'


FILE: 'MAIN.CML'

$INCLUDE: 'CASE.CMN'
$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CONTRL.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'SWEEP.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'NRMLBC.CML'

$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CVLCTY.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'NMWSH.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLCRD.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'PNLVEC.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'OUT1.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLCRD.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'
```

```
FILE: 'OUT2.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKCRD.CMN'
$INCLUDE: 'WAKNDX.CMN'


FILE: 'OUT3.CML'

$INCLUDE: 'ALSLP.CMN'
$INCLUDE: 'BELOAD.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CALPHA.CMN'
$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CVLCTY.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'


FILE: 'OUT4.CML'

$INCLUDE: 'BLLOAD.CMN'
$INCLUDE: 'CONSTS.CMN'


FILE: 'OUT5.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'RTLOAD.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'OUT6.CML'

$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WRATES.CMN'


FILE: 'OUT8.CML'

$INCLUDE: 'UNITS.CMN'
```

TABLE B-4 (CONTINUED)

```
FILE: 'PREWAK.CML'

$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKNDX.CMN'


FILE: 'PSURF.CML'

$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CALPHA.CMN'
$INCLUDE: 'CONFIG.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PNLCRD.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'PNLVEC.CMN'
$INCLUDE: 'REFSUR.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'PWAKE.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKCRD.CMN'
$INCLUDE: 'WAKNDX.CMN'


 ILE: 'READAF.CML'
$INCLUDE: 'AFTABL.CMN'
$INCLUDE: 'UNITS.CMN'
```

```
FILE: 'READIN.CML'

$INCLUDE: 'CASE.CMN'
$INCLUDE: 'CGRID.CMN'
$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONFIG.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CONTRL.CMN'
$INCLUDE: 'CTCON.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'SWEEP.CMN'
$INCLUDE: 'UNITS.CMN'
$INCLUDE: 'WAKNDX.CMN'


FILE: 'REDTAB.CML'

$INCLUDE: 'UNITS.CMN'


FILE: 'RFSURF.CML'

$INCLUDE: 'AFTABL.CMN'
$INCLUDE: 'BLDELE.CMN'
$INCLUDE: 'CALPHA.CMN'
$INCLUDE: 'CGRID.CMN'
$INCLUDE: 'CONFIG.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'PNLVEC.CMN'
$INCLUDE: 'REFSUR.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'ROLNDX.CML'

$INCLUDE: 'CCIRC.CMN'
$INCLUDE: 'CONTRL.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PNLNDX.CMN'
$INCLUDE: 'WAKNDX.CMN'
```

TABLE B-4 (CONTINUED)

```
FILE: 'TIMER.CML'

$INCLUDE: 'CTIMER.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'TRACE.CML'

$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'UNITS.CMN'


FILE: 'VCYLN.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'GVINT.CMN'
$INCLUDE: 'WRATES.CMN'


FILE: 'VLINE.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'GVINT.CMN'


FILE: 'WAKPRM.CML'

$INCLUDE:FILE: 'CTCON.CMN'

$INCLUdE: 'CWIND.CMN'
$INCLUDE: 'PARMS.CMN'
$INCLUDE: 'WRATES.CMN'


FILE: 'WAKUPD.CML'

$INCLUDE: 'CIPRT.CMN'
$INCLUDE: 'CONTRL.CMN'
$INCLUDE: 'CTCON.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'WRATES.CMN'


FILE: 'WNDPRO.CML'

$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'CWIND.CMN'
$INCLUDE: 'IFLAGS.CMN'
$INCLUDE: 'PARMS.CMN'
```

TABLE B-4 (CONCLUDED)

```
FILE: 'ZROLFT.CML'

$INCLUDE: 'AFTABL.CMN'
$INCLUDE: 'CONSTS.CMN'
$INCLUDE: 'UNITS.CMN'
```

APPENDIX C

PROGRAM LSWT SOURCE LISTING

```
C     ******************************************************************00000010
C     ******************************************************************00000020
C                                                                        00000030
C     LSWT VERSION 1.00  RELEASED MAY 1986                               00000040
C                                                                        00000050
C     HORIZONTAL AXIS WIND TURBINE AERODYNAMIC PERFORMANCE ANALYSIS      00000060
C     BASED ON LIFTING SURFACE THEORY WITH PRESCRIBED WAKE               00000070
C                                                                        00000080
C     TECHNICAL DEVELOPMENT AND PROGRAMMING BY                           00000090
C     J. DAVID KOCUREK                                                   00000100
C     COMPUTATIONAL METHODOLOGY ASSOCIATES                               00000110
C                                                                        00000120
C     DEVELOPED UNDER MICROSOFT FORTRAN77, VERSION 3.20 (MS-DOS)         00000130
C                                                                        00000140
C     ******************************************************************00000150
C     ******************************************************************00000160
```

```
C     ******************************************************************
C
C     LSWT01(MAIN)
C
C     ******************************************************************
C
      PROGRAM MAIN
C
      COMMON/ CASE   / IPOINT, ALPHAD, VWOD,   IVDIM,  VWFAC,  PSIRD,
     1                 CASEID
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ SWEEP  / ALPSIN, DALP,   NALP,   VWOIN,  DVWO,   NVWO,
     1                 PSIRIN, DPSIR,  NPSIR
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     OPEN DISK I/O FILES
C
      CALL IOINIT
C
C     INITIALIZE START TIME
C
      CALL TIMER(1)
      CALL TIMER(3)
C
C     CALL INITIALIZATION AND INPUT EXECUTIVE SUBROUTINE
C
      CALL EXINT
      IF(IPRT7.GT.0) CALL TRACE('EXINT ')
C
      IPOINT = 0
C
C     --------------------------------
C     LOOP 7000 SWEEPS GEOMETRIC PITCH
```

```
C     -------------------------------
C
      DO 7000   IALP = 1,NALP
C
      ALPHAS = ALPSIN+(IALP-1)*DALP
      ALPHAD = ALPHAS/RPD
C
C     --------------------------
C     LOOP 6000 SWEEPS WIND SPEED
C     --------------------------
C
      DO 6000  IVWO = 1,NVWO
C
      IPOINT = IPOINT + 1
      VWO = VWOIN+(IVWO-1)*DVWO
      IF(IVDIM.EQ.3) VWO = 1.0/VWO/VWFAC
      VWOD = VWO*VWFAC
      IF(IVDIM.EQ.3) VWOD = 1.0 / VWOD
      IF( (NVWO.EQ.1) .AND. (IALP.GT.1) ) GO TO 110
C
C     CALL BLADE GEOMETRY EXECUTIVE SUBROUTINE
C
      CALL EXBLD
      IF(IPRT7.GT.0) CALL TRACE('EXBLD ')
C
      IF(IPOINT.GT.1) GO TO 110
C
C     REFERENCE BLADE GEOMETRY REPORT, IPRT = 3, IPOINT = 1
C
      IF(IPRT1.EQ.3) CALL PSURF(0.0,0.0,1)
      IF(IPRT1.EQ.3) CALL OUT1(0)
C
C     GET INITIAL LIFT SLOPE (FIRST POINT ONLY)
C
      CALL INTLS(ALPHAS)
C
C     INITIALIZE CIRCULATION ARRAYS
C
      DO 100  J = 1, NPANEL
      CIRC(J) = 0.0
      CIRCR(J) = 1.0
  100 CONTINUE
C
  110 CONTINUE
C
C     ------------------------------------------------------------
C     LOOP 5000 SWEEPS AZIMUTH FOR QUASI-STEADY CASE, ELSE ONE POINT
C     ------------------------------------------------------------
C
      IF(ISTDY.EQ.1) GO TO 200
C
      PSIRIN = 0.0
```

```
      DPSIR  = 0.0
      NPSIR  = 1
C
  200 CONTINUE
C
      DO 5000   IPSIR = 1,NPSIR
C
      PSIR = PSIRIN+(IPSIR-1)*DPSIR
      PSIRD = PSIR/RPD
C
      IF((IPRT1.NE.1) .AND. (IPRT1.NE.2)) GO TO 400
      IF((IPRT1.EQ.2) .AND. (IPSIR.GT.1)) GO TO 400
C
C     BLADE GEOMETRY REPORT, IPRT = 1,2
C
      PSIBS = 2.0*PI/NBLDS
C
      DO 300 ITBLD = 1, NBLDS
      PSITB = PSIR-(ITBLD-1)*PSIBS
      CALL PSURF(ALPHAS,PSITB,1)
      CALL OUT1(ITBLD)
  300 CONTINUE
C
  400 CONTINUE
C
C     SET SWITCH SO SURFACE AIC'S WILL BE CALCULATED ONLY ONCE
C     PER AZIMUTH
C
      IAICDS = 0
C
C     ----------------------------------
C     LOOP 4000 CONVERGES PRESCRIBED WAKE
C     ----------------------------------
C
      IWCYLM = ICYL3
      IF(ICYL3.EQ.0) IWCYLM = 6
C
      DO 4000   IWCYL = 1, IWCYLM
C
C     ---------------------------------
C     LOOP 3000 CONVERGES ROLL-UP OPTION
C     ---------------------------------
C
      IRCYLM = ICYL4
      IF(ICYL4.EQ.0) IRCYLM = 4
C
      DO 3000  IRCYL = 1,IRCYLM
C
C     ------------------------------------------------
C     LOOP 2000 SWEEPS TARGET BLADES, BLADE ONE AT PSIR
C     ------------------------------------------------
C
```

```
      ITBLDM = NBLDS
      IF(ISTDY.EQ.0) ITBLDM = 1
C
C     CLEAR OUTPUT BUFFER
C
      REWIND IOU8
C
      DO 2000 ITBLD = 1,ITBLDM
C
C     CALL AERODYNAMIC INFLUENCE COEFFICIENT EXECUTIVE
C
      CALL EXAIC(ALPHAS,PSIR,ITBLD,IAICDS)
      IF(IPRT7.GT.0) CALL TRACE('EXAIC ')
C
C     RESET SWITCH SO SURFACE AIC'S WILL NOT BE RECALCULATED
C
      IAICDS = 1
C
C     ----------------------------------------------------
C     LOOP 1000 CONVERGES EFFECTIVE LINEAR LIFT CURVE SLOPE
C     ----------------------------------------------------
C
      ISCYLM = ICYL2
      IF(ICYL2.EQ.0) ISCYLM = 4
C
      DO 1000  ISCYL = 1,ISCYLM
C
C     CALL SOLUTION EXECUTIVE
C
      CALL EXSLV(ITBLD,PSIR,IPSIR,ISCYL)
      IF(IPRT7.GT.0) CALL TRACE('EXSLV ')
C
C     CHECK FOR CONVERGENCE OF LINEAR LIFT CURVE SLOPES
C
      IF(LSFLAG.EQ.0) GO TO 1500
      IF(IWFLAG.EQ.1) GO TO 1500
      IF(IRFLAG.EQ.1) GO TO 1500
C
C     --------
 1000 CONTINUE
C     --------
C
 1500 CONTINUE
C
C     REPORT WAKE PARAMETERS, IPRT6 = 1
C
      IF(IPRT6.EQ.1) CALL OUT6
C
C     REPORT WAKE COORDINATES, IPRT2=1
C
      IF(IPRT2.EQ.1) CALL OUT2(ITBLD,PSIR)
C
```

```
C     BLADE ELEMENT PERFORMANCE SUMMARY,IPRT3 = 1
C
      IF(IPRT3.EQ.1) CALL OUT3(ITBLD,IOU6)
C
C     BLADE PERFORMANCE SUMMARY, IPRT4 = 1
C
      IF(IPRT4.EQ.1) CALL OUT4(ITBLD,IOU6)
C
C     ROTOR SUMMARY PERFORMANCE, IPRT5 = 1
C
      IF(IPRT5.EQ.1) CALL OUT5
C
C     LOAD OUTPUT BUFFER AS REQUESTED
C
      IF((IPRT3.EQ.2) .OR. (IPRT3.EQ.3)) CALL OUT3(ITBLD,IOU8)
      IF((IPRT4.EQ.2) .OR. (IPRT4.EQ.3)) CALL OUT4(ITBLD,IOU8)
C
      IF(ISTDY.EQ.0) GO TO 2000
C
C     UPDATE AZIMUTH AVERAGE FOR ITBLD'S NEW CIRCULATION
C
      DO 1600  J = 1,NPANEL
      IF(ITBLD.EQ.1) ACIRC(J) = CIRC(J)/NBLDS
      IF(ITBLD.GT.1) ACIRC(J) = ACIRC(J)+CIRC(J)/NBLDS
 1600 CONTINUE
C
C     --------
 2000 CONTINUE
C     --------
C
C     CHECK FOR CONVERGENCE OF ROLL-UP
C
      IF(IWFLAG.EQ.1) GO TO 3500
      IF(IRFLAG.EQ.0) GO TO 3500
C
C     --------
 3000 CONTINUE
C     --------
C
 3500 CONTINUE
C
C     CHECK FOR CONVERGENCE OF WAKE
C
      IF(IWFLAG.EQ.0) GO TO 4100
C
C     --------
 4000 CONTINUE
C     --------
C
 4100 CONTINUE
C
C     REPORT WAKE PARAMETERS,IPRT6 = 2
```

```
C
      IF(IPRT6.EQ.2) CALL OUT6
C
C     REPORT WAKE COORDINATES, IPRT2 = 2
C
      IF(IPRT2.NE.2) GO TO 4300
C
      ITBLDM = NBLDS
      IF(ISTDY.EQ.0) ITBLDM = 1
      DO 4200  ITBLD = 1,ITBLDM
      IF(IPRT2.EQ.2) CALL OUT2(ITBLD,PSIR)
 4200 CONTINUE
C
 4300 CONTINUE
C
C     BLADE ELEMENT AND/OR BLADE SUMMARY PERFORMANCE,
C     IPRT3 = 2, AND/OR IPRT4 = 2 (RESTORE FROM BUFFER)
C
      IF((IPRT3.EQ.2) .OR. (IPRT4.EQ.2)) CALL APPEND(IOU8,IOU6,132,0)
C
C     ROTOR SUMMARY PERFORMANCE, IPRT5 = 2
C
      IF(IPRT5.EQ.2) CALL OUT5
C
      IF(ISTDY.EQ.0) GO TO 4500
C
C     UPDATE DISK AVERAGE CIRCULATION FOR THIS AZIMUTH
C
      DO 4400 J = 1, NPANEL
      IF(IPSIR.EQ.1) DCIRC(J) = ACIRC(J)/IPSIR
      IF(IPSIR.GT.1) DCIRC(J) = (DCIRC(J)*(IPSIR-1)+ACIRC(J))/IPSIR
 4400 CONTINUE
C
 4500 CONTINUE
C
C     --------
 5000 CONTINUE
C     --------
C
C     REPORT WAKE PARAMETERS,IPRT6 = 3
C
      IF(IPRT6.EQ.3) CALL OUT6
C
C     REPORT WAKE COORDINATES, IPRT2 = 3
C
      IF(IPRT2.NE.3) GO TO 5200
C
      ITBLDM = NBLDS
      IF(ISTDY.EQ.0) ITBLDM = 1
      DO 5100  ITBLD = 1,ITBLDM
      IF(IPRT2.EQ.3) CALL OUT2(ITBLD,PSIR)
 5100 CONTINUE
```

```
C
 5200 CONTINUE
C
C     BLADE ELEMENT AND/OR BLADE SUMMARY PERFORMANCE,
C     IPRT3 = 3, AND/OR IPRT4 = 3 (RESTORE FROM BUFFER)
C
      IF((IPRT3.EQ.3) .OR. (IPRT4.EQ.3)) CALL APPEND(IOU8,IOU6,132,0)
C
C     ROTOR SUMMARY PERFORMANCE, IPRT5 = 3
C
      IF(IPRT5.EQ.3) CALL OUT5
C
C     CONVERGENCE MESSAGES FOR THIS POINT
C
      IF(LSFLAG.EQ.1) WRITE(IOU6,1001) ISCYLM
 1001 FORMAT('0',4X,41('*')/
     1       ' ',4X,'LIFT SLOPES NOT CONVERGED AFTER ',I2,' CYCLES'/
     2       ' ',4X,41('*'))
C
      IF(IRFLAG.EQ.1) WRITE(IOU6,3001) IRCYLM
 3001 FORMAT('0',4X,41('*')/
     1       ' ',4X,'ROLL-UP NOT CONVERGED AFTER ',I2,' ITERATIONS'/
     2       ' ',4X,41('*'))
C
      IF(IWFLAG.EQ.1) WRITE(IOU6,4001) IWCYLM
 4001 FORMAT('0',4X,38('*')/
     1       ' ',4X,'WAKE NOT CONVERGED AFTER ',I2,' ITERATIONS'/
     2       ' ',4X,38('*'))
C
C     --------
 6000 CONTINUE
C     --------
C
C     --------
 7000 CONTINUE
C     --------
C
C     DISPLAY TOTAL TIME
C
      CALL TIMER(2)
C
C     CLOSE DISK I/O FILES
C
      CALL IOFNSH
C
      STOP ' '
      END
```

```
C     ******************************************************************
C
C     LSWT01(INTLS)
C
C     CALCULATES THE INITIAL ESTIMATE OF THE EFFECTIVE LINEAR LIFT CURVE
C     SLOPE AT EACH BLADE SOLUTION STATION
C
C     ******************************************************************
C
      SUBROUTINE INTLS(ALPHAS)
C
      COMMON/ AFTABL / NAFT,   IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
      COMMON/ ALSLP  / ALS(20),        ALSNEW(20),     ALSOLD(20)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),
     2                 TWSTCG(20)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
C
C     ******************************************************************
C
C     INITIAL VALUE OF LINEAR LIFT SLOPE
C
      DO 200   J = 1,NP
C
      VEL = SQRT(VWO**2+(OMEGAR*RBE(J))**2)
      REN = RHO*VEL*CHDCGL(J)*RADIUS/AMU
C
      ALTAB = ALPHAS/2.0-AZLLC(J)+ATAN(VWO/OMEGAR/RBE(J))
      IF (ABS(AZLLC(J)).LT.1.0E-04.AND.ABS(ALTAB).LT.1.0E-04)
     1   ALTAB = AZLLC(J)/RPD + 1.0
      CALL AFCOEF(J,ALTAB,REN,CL,CD,CM,INOREX)
C
```

```
C     INITIAL ESTIMATE OF EFFECTIVE LINEAR LIFT SLOPE USING LEADING     00000530
C     EDGE ANGLE TO CORRECT FOR SWEEP - LATER UPDATES WILL USE          00000540
C     AERODYNAMIC SWEEP                                                 00000550
C                                                                       00000560
      ALTAB  = ALTAB*RPD                                                00000570
      ALS(J) = CL/(ALTAB-AZLLC(J)) * CLAMLE(J)                          00000580
C                                                                       00000590
  200 CONTINUE                                                          00000600
C                                                                       00000610
      RETURN                                                            00000620
      END                                                               00000630
```

```
C     ******************************************************************
C
C     LSWT01(EXINT)
C
C     PERFORMS INITIALIZATION OF CONSANTS, CALLS INPUT SUBROUTINE, AND
C     REPORTS RECORD IMAGE OF INPUT
C
C     ******************************************************************
C
      SUBROUTINE EXINT
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONSTS / PI,     RPD
C
C     ******************************************************************
C
      PI = ACOS(-1.0)
      RPD = PI/180.0
C
C     CALL INPUT SUBROUTINE
C
      CALL READIN
      IF(IPRT7.GT.0) CALL TRACE('READIN')
C
C     OUTPUT RECORD IMAGE OF INPUT, IPRT8 = 1
C
      IF(IPRT8.EQ.1) CALL OUT8
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT01(EXBLD)
C
C     BLADE GEOMETRY EXECUTIVE
C
C     ******************************************************************
C
      SUBROUTINE EXBLD
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
C
C     ******************************************************************
C
C     GENERATE NONDIMENSIONAL PANEL GRID AND
C     COLLOCATION POINTS
C
      CALL GRID
      IF(IPRT7.GT.0) CALL TRACE('GRID  ')
C
C     CALCULATE TWO-DIMENSIONAL AIC'S
C
      CALL AICDLS
      IF(IPRT7.GT.0) CALL TRACE('AICDLS')
C
C     GENERATE REFERENCE SURFACE PANEL COORDINATES
C
      CALL RFSURF
      IF(IPRT7.GT.0) CALL TRACE('RFSURF')
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT01(EXAIC)
C
C     AERODYNAMIC INFLUENCE COEFFICIENT EXECUTIVE
C
C     ******************************************************************
C
      SUBROUTINE EXAIC(ALPHAS,PSIR,ITBLD,IAICDS)
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONSTS / PI,     RPD
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
C
C     ******************************************************************
C
C     AZIMUTH OF TARGET BLADE
C
      PSIBS = 2.0*PI/NBLDS
      PSITB = PSIR-(ITBLD-1)*PSIBS
C
C     SWEEP SOURCE BLADES
C
      DO 200   ISBLD = 1,NBLDS
C
C     AZIMUTH OF SOURCE BLADE
C
      PSISB = PSIR-(ISBLD-1)*PSIBS
C
C     POSITION SURFACE
C
      CALL PSURF(ALPHAS,PSISB,ISBLD)
      IF(IPRT7.GT.0) CALL TRACE('PSURF ')
C
C     SURFACE INFLUENCE COEFFICIENTS OF SOURCE ON TARGET
C     (THESE ARE CALCULATED ONLY ONCE PER CASE SINCE RELATIVE
C     POSITION OF BLADES TO EACH OTHER IS FIXED)
C
      IF(IAICDS.EQ.0) CALL AICDS(ISBLD)
      IF((IAICDS.EQ.0) .AND. (IPRT7.GT.0)) CALL TRACE('AICDS ')
C
C     CALCULATE WAKE GEOMETRY OF SOURCE BLADE
C
      IF(ISTDY.EQ.0.AND.ISBLD.GT.1) GO TO 100
C
      CALL PREWAK(PSISB,ISBLD)
      IF(IPRT7.GT.0) CALL TRACE('PREWAK')
```

```
C
      IF(ISTDY.EQ.0) CALL PWAKE
      IF((ISTDY.EQ.0) .AND. (IPRT7.GT.0)) CALL TRACE('PWAKE ')
C
  100 CONTINUE
C
C     WAKE INFLUENCE COEFFICIENTS OF SOURCE ON TARGET
C
      CALL AICDW(ISBLD)
      IF(IPRT7.GT.0) CALL TRACE('AICDW ')
C
  200 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT01(EXSLV)
C
C     CIRCULATION SOLUTION EXECUTIVE
C
C     ******************************************************************
C
      SUBROUTINE EXSLV(ITBLD,PSIR,IPSIR,ISCYL)
C
      COMMON/ CASE   / IPOINT, ALPHAD, VWOD,   IVDIM,  VWFAC,  PSIRD,
     1                 CASEID
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
C
C     ******************************************************************
C
C     CALCULATE NORMAL BOUNDARY CONDITION
C
      PSIT = PSIR-(2.0*PI)/NBLDS*(ITBLD-1)
C
      CALL NRMLBC(PSIT)
      IF(IPRT7.GT.0) CALL TRACE('NRMLBC')
C
      IF(ISTDY.EQ.0) GOTO 200
C
C     PREPARE FOR OUTER SOLUTION OF CIRCULATION
C
      TOLMAX = 1.0E-05
      IF(TOL1.GT.0.0) TOLMAX = TOL1
      TOLMAX = TOLMAX*100.0
C
C     INITIALIZE CONVERGENCE CHECK
C
      SUMCR = 0.0
C
C     ALLOW ITCYLM CYLES TO CONVERGE
```

```
C
      ITCYLM = ICYL1
      IF(ITCYLM.EQ.0) ITCYLM = 100
      ITCYLM = ITCYLM/20
C
C     INITIALIZE CIRCULATION RATIO
C
      DO 100 J = 1, NPANEL
      CIRCR(J) = 1.0
  100 CONTINUE
C
  200 CONTINUE
C
C     LOOP 1000 CONVERGES CIRCULATION OUTER SOLUTION
C
      IF(ISTDY.EQ.0) ITCYLM = 1
C
      DO 1000 ITCYL = 1, ITCYLM
C
C     ASSEMBLE INFLUENCE COEFFICIENT MATRIX
C
      CALL AICDSW(ITBLD)
      IF(IPRT7.GT.0) CALL TRACE('AICDSW')
C
C     SOLVE LINEAR EQUATION SYSTEM BY GAUSS-SEIDEL ITERATION
C
      CALL GSITER(NPANEL)
      IF(IPRT7.GT.0) CALL TRACE('GSITER')
C
      IF(ISTDY.EQ.0) GO TO 1100
C
      IF(IPOINT*IPSIR.EQ.1) GO TO 1100
C
C     UPDATE CIRCULATION RATIO PARAMETER
C
      SUMCRP = 0.0
      DO 300   J = 1, NPANEL
      CIRCRI = CIRCR(J)
      CIRCRO = DCIRC(J)/CIRC(J)
      CIRCR(J) = 0.5*(CIRCRO-CIRCRI)+CIRCRI
      SUMCRP=SUMCRP + ABS(CIRCRO)
  300 CONTINUE
C
C     CHECK CONVERGENCE
C
      TOL = ABS((SUMCRP-SUMCR)/SUMCRP)
      IF(TOL.LE.TOLMAX) GO TO 1100
C
      SUMCR = SUMCRP
C
 1000 CONTINUE
C
```

```
 1100 CONTINUE
C
C     CALCULATE CIRCULATION PARAMETERS
C
      CALL CIRPRM
      IF(IPRT7.GT.0) CALL TRACE('CIRPRM')
C
C     IF OPTIONED, CHECK ROLL-UP COMPATABILITY WITH LOADING
C
      IRFLAG = 0
      IF(IROLL.EQ.1) CALL ROLNDX
      IF((IROLL.EQ.1) .AND. (IPRT7.GT.0)) CALL TRACE('ROLNDX')
C
C     CALCULATE BLADE ELEMENT VELOCITIES
C
      CALL BEVEL(PSIT)
      IF(IPRT7.GT.0) CALL TRACE('BEVEL ')
C
C     CALCULATE BLADE ELEMENT ALPHAS
C
      CALL BEALPH
      IF(IPRT7.GT.0) CALL TRACE('BEALPH')
C
C     CALCULATE BLADE AND ROTOR PERFORMANCE
C
      CALL AEROPF(ITBLD,IPSIR,ISCYL)
      IF(IPRT7.GT.0) CALL TRACE('AEROPF')
C
C     CHECK FOR CONVERGENCE OF WAKE PARAMETERS
C
      CALL WAKUPD
      IF(IPRT7.GT.0) CALL TRACE('WAKUPD')
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWTO1(BVTINT)
C
C     PERFORMS A TABLE LOOKUP OF Y=F(X) OR Y=F(X,Z) USING BISECTION
C     ITERATION TO LOCATE TARGETS.  LINEAR INTERPOLATION IS USED TO
C     CALCULATE Y FROM TABLE MEMBERS.
C
C     RETURNS INOREX TO INDICATE TARGET POSITION IN X RANGE IF Y=F(X),
C     OR IN Z RANGE IF Y=F(X,Z).
C
C            INOREX = -1  IF EXTRAPOLATION BELOW RANGE
C                     0  IF INTERPOLATION WITHIN RANGE
C                     1  IF EXTRAPOLATION ABOVE RANGE
C
C     ******************************************************************
C
      SUBROUTINE BVTINT(X,Y,Z,NX,NZ,TABLE,INOREX)
C
      DIMENSION TABLE(1)
C
C     ******************************************************************
C
C     FIRST PASS IS TO LOCATE X TARGET POSITION AMONG TABLE MEMBERS,
C     SECOND PASS IS FOR Z TARGET IF REQUIRED.
C
      IPASS  = 1
      TARGET = X
      NMAX   = NX
      NL     = 1
      NH     = NX
      NLOW   = NL
      NMID   = NMAX/2
      NHIGH  = NMAX
C
  100 CONTINUE
      IF(NMAX.LE.2) GO TO 500
C
C     DETERMINE WHICH TABLE HALF TARGET IS IN
C
  110 CONTINUE
      IF(TARGET-TABLE(NMID)) 200,300,400
C
  200 CONTINUE
C
C     TARGET IS BELOW MIDPOINT MEMBER
C
      NHALF = (NMID-NLOW)/2
      IF(NHALF) 220,220,210
  210 CONTINUE
      NHIGH = NMID
      NMID  = NLOW+NHALF
```

```
      GO TO 110
  220 CONTINUE
      NL = NLOW
      NH = NMID
      GO TO 500
C
C     TARGET EQUAL TO MIDPOINT MEMBER
C
  300 CONTINUE
      NL = NMID-1
      NH = NMID
      GO TO 500
C
  400 CONTINUE
C
C     TARGET IS ABOVE MIDPOINT MEMBER
C
      NHALF = (NHIGH-NMID)/2
      IF(NHALF) 420,420,410
  410 CONTINUE
      NLOW = NMID
      NMID = NLOW+NHALF
      GO TO 110
  420 CONTINUE
      NL = NMID
      NH = NHIGH
C
  500 CONTINUE
C
C     TARGET LOCATED
C
      GO TO (600,700), IPASS
C
  600 CONTINUE
      NLX = NL
      NHX = NH
      NLZ = NX+1
C
C     RESET TO FIND Z POSITION IF REQUIRED
C
      IF(NZ.LE.0) GO TO 800
      IPASS  = 2
      TARGET =  Z
      NMAX   = NZ
      NL     = NX+1
      NH     = NX+NZ
      NLOW   = NL
      NMID   = NLOW+NMAX/2
      NHIGH  = NX+NMAX
      GO TO 100
C
  700 CONTINUE
```

```
      NLZ = NL
      NHZ = NH
C
  800 CONTINUE
C
C     INTERPOLATE IN X AT LOW Z POSITION
C
      XL   = TABLE(NLX)
      XH   = TABLE(NHX)
      NTAB = (NX+NZ)+(NLZ-NX-1)*NX+NLX
      YL   = TABLE(NTAB)
      YH   = TABLE(NTAB+1)
      IF(NX.EQ.1) YLOW = YL
      IF(NX.GT.1) YLOW = ((YH-YL)/(XH-XL))*(X-XL)+YL
C
C     SET INOREX FOR X RANGE
C
      INOREX = 0
      IF(X.LT.XL) INORX = -1
      IF(X.GT.XH) INORX = 1
C
C     RETURN IF Y = F(X) ONLY
C
      Y = YLOW
      IF(NZ.LE.0) GO TO 900
C
C     INTERPOLATE IN X AT HIGH Z POSITION
C
      ZL   = TABLE(NLZ)
      ZH   = TABLE(NHZ)
      NTAB = NTAB+NX
      YL   = TABLE(NTAB)
      YH   = TABLE(NTAB+1)
      IF(NX.EQ.1) YHIGH = YL
      IF(NX.GT.1) YHIGH = ((YH-YL)/(XH-XL))*(X-XL)+YL
C
C     INTERPOLATE IN Z
C
      IF(NZ.EQ.1) Y = YLOW
      IF(NZ.GT.1) Y = (YHIGH-YLOW)/(ZH-ZL)*(Z-ZL)+YLOW
C
C     RESET INOREX FOR Z RANGE
C
      INOREX = 0
      IF(Z.LT.ZL) INOREX=-1
      IF(Z.GT.ZH) INOREX=1
C
  900 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT01(ZROLFT)
C
C     USES BISECTION ITERATION TO LOCATE THE ZERO LIFT LINE ANGLE AT
C     A SPECIFIED REYNOLDS NUMBER FROM BIVARIENT AIRFOIL TABLES.  IF
C     MUTIPLE TABLES DEFINE AIRFOIL THEN THE RESULT IS AVERAGED BETWEEN
C     TABLES.
C
C     ******************************************************************
C
      SUBROUTINE ZROLFT(ALPO,RE,JTAB1,JTAB2,AFTWF)
C
      COMMON/ AFTABL / NAFT,   IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     FIRST PASS IS FOR JTAB1 AIRFOIL, SECOND PASS FOR JTAB2 IF REQUIRED
C
      JTAB = JTAB1
      IPASS = 1
C
   50 CONTINUE
C
C     SET INITIAL GUESS TO ZERO AND SENSE DIRECTION TO MARCH FOR TRAP
C
      ALPO = 0.0
      CALL BVTINT(RE,CL,ALPO,NRCL(JTAB),NACL(JTAB),CLTAB(1,JTAB),INOREX)
      IF(CL) 100,500,200
C
  100 CONTINUE
C
C     SEARCH IN POSITIVE DIRECTION ON ALPHA
C
      AL1 = ALPO
      DO 120   ICOUNT = 1,20
      AL2 = AL1+1.0
      ALPO = AL2
      CALL BVTINT(RE,CL,ALPO,NRCL(JTAB),NACL(JTAB),CLTAB(1,JTAB),INOREX)
      IF(CL) 110,500,300
  110 CONTINUE
      AL1 = AL2
  120 CONTINUE
      GO TO 450
```

```
C
  200 CONTINUE
C
C     SEARCH IN NEGATIVE DIRECTION ON ALPHA
C
      AL2 = ALPO
      DO 220   ICOUNT = 1,20
      AL1 = AL2-1.0
      ALPO = AL1
      CALL BVTINT(RE,CL,ALPO,NRCL(JTAB),NACL(JTAB),CLTAB(1,JTAB),INOREX)
      IF(CL) 300,500,210
  210 CONTINUE
      AL2 = AL1
  220 CONTINUE
      GO TO 450
C
  300 CONTINUE
C
C     ALPHA FOR ZERO LIFT HAS BEEN TRAPPED
C
      DO 400   ICOUNT = 1,20
      ALPO = (AL1+AL2)/2.0
      CALL BVTINT(RE,CL,ALPO,NRCL(JTAB),NACL(JTAB),CLTAB(1,JTAB),INOREX)
      IF(ABS(CL).LE.1.0E-04) GO TO 500
      IF(CL) 410,500,420
  410 CONTINUE
      AL1 = ALPO
      GO TO 400
  420 CONTINUE
      AL2 = ALPO
  400 CONTINUE
C
  450 CONTINUE
C
C     REPORT FAILURE AND HALT EXECUTION
C
      WRITE(IOU6,10)
   10 FORMAT('0',4X,49('*')/
     1       ' ',4X,'FAILURE TO FIND ZERO LIFT LINE...EXECUTION HALTED'/
     2       ' ',4X,49('*'))
C
      STOP ' '
C
  500 CONTINUE
C
C     ZERO LIFT ALPHA FOUND FOR THIS PASS, RESET PARAMETERS AND REPEAT
C     FOR SECOND TABLE IF REQUIRED
C
      IF(JTAB1.EQ.JTAB2) GO TO 700
      IF(IPASS.EQ.2)     GO TO 600
      IPASS = 2
      ALPO1 = ALPO
```

```
      JTAB = JTAB2                                                      00001050
      GO TO 50                                                          00001060
C                                                                       00001070
  600 CONTINUE                                                          00001080
C                                                                       00001090
C     ZERO LIFT LINE AVERAGED BETWEEN TABLES                            00001100
C                                                                       00001110
      ALPO = ALPO*AFTWF+ALPO1*(1.0-AFTWF)                               00001120
C                                                                       00001130
  700 CONTINUE                                                          00001140
      ALPO = ALPO*RPD                                                   00001150
C                                                                       00001160
      RETURN                                                            00001170
      END                                                               00001180
```

```
C     ******************************************************************
C
C     LSWTO2(AICDLS)
C
C     CALCULATES TWO-DIMENSIONAL SELF-INDUCED INFLUENCE COEFFICIENTS
C
C     ******************************************************************
C
      SUBROUTINE AICDLS
C
      COMMON/ CDLS   / DLS(25)
      COMMON/ CGRID  / NGRID(4),       XG(6),  YG(24), XCG(5)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
C
C     ******************************************************************
C
      OO2PI = 1.0/2.0/PI
C
C     SELECT COLLOCATION POINT ON SEGMENT
C
      DO 200 M = 1, MP
C
C     SWEEP VORTEX FILAMENTS ON SURFACE
C
      DO 100 I = 1, MP
      DXL = XCG(M)-XG(I)
      DXT = XCG(M)-XG(I+1)
      ISTR = (I-1)*MP+M
      DLS(ISTR) = -OO2PI*(1.0/DXL-1.0/DXT)
  100 CONTINUE
C
C     WAKE FILAMENT
C
      DW = -OO2PI/DXT
      ISTR = MP**2
      DLS(ISTR) = DLS(ISTR)+TECF*DW
  200 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWTO2(AICDS)
C
C     CALCULATES THE INFLUENCE COEFFICIENTS DUE TO THE LIFTING SURFACE
C
C     ******************************************************************
C
      SUBROUTINE AICDS(IBLD)
C
      DIMENSION  DSN(100),DST(100)
C
      COMMON/ CONSTS / PI,     RPD
      COMMON/ GVINT  / XA,     YA,     ZA,     XB,     YB,     ZB,
     1                 XC,     YC,     ZC,     AN1C,   AN2C,   AN3C,
     2                 AT1C,   AT2C,   AT3C
      COMMON/ PNLCRD / XS(6,24),       YS(6,24),       ZS(6,24),
     1                 XCOL(5,20),     YCOL(5,20),     ZCOL(5,20)
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ PNLVEC / AN1(5,20),      AN2(5,20),      AN3(5,20),
     1                 AT1(5,20),      AT2(5,20),      AT3(5,20),
     2                 AQC1(20),       AQC2(20),       AQC3(20)
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     SELECT COLLOCATION POINT
C
      DO 400   M = 1,MP
      DO 400   N = 1,NP
      LP = (M-1)*NP+N
      XC = XCOL(M,N)
      YC = YCOL(M,N)
      ZC = ZCOL(M,N)
      AN1C = AN1(M,N)
      AN2C = AN2(M,N)
      AN3C = AN3(M,N)
      AT1C = AT1(M,N)
      AT2C = AT2(M,N)
      AT3C = AT3(M,N)
C
C     INITIALIZE ROW OF COEFFICIENTS
C
      DO 100   I = 1,NPANEL
      DSN(I) = 0.0
      DST(I) = 0.0
  100 CONTINUE
C
```

```
C     SWEEP ALONG SPANWISE ELEMENTS IN CHORDWISE SEQUENCE
C
      DO 200   I = 1,MP1
C
      DO 200   JSEG = 1,NSGMNT
      JG1  = JGLIM(JSEG,1)
      JG2  = JG1+NPS(JSEG)-1
      JCG1 = JCGLIM(JSEG,1)
      JCG2 = JCGLIM(JSEG,2)
C
      DO 200   J = JG1,JG2
      JCG = JCG1 + (J-JG1)
      XA  = XS(I,J)
      YA  = YS(I,J)
      ZA  = ZS(I,J)
      JP1 = J + 1
      XB  = XS(I,JP1)
      YB  = YS(I,JP1)
      ZB  = ZS(I,JP1)
C
      CALL VLINE(AIN,AIT)
C
      KPP = (I-1)*NP + JCG
      KPM = (I-2)*NP + JCG
      IF(I.LT.MP1)  DSN(KPP) = DSN(KPP) + AIN
      IF(I.GT.1)    DSN(KPM) = DSN(KPM) - AIN
      IF(I.LT.MP1)  DST(KPP) = DST(KPP) + AIT
      IF(I.GT.1)    DST(KPM) = DST(KPM) - AIT
C
  200 CONTINUE
C
C     SWEEP ALONG CHORDWISE FILAMENTS IN SPANWISE SEQUENCE
C
      DO 300   JSEG = 1,NSGMNT
      JG1  = JGLIM(JSEG,1)
      JG2  = JGLIM(JSEG,2)
      JCG1 = JCGLIM(JSEG,1)
      JCG2 = JCGLIM(JSEG,2)
C
      DO 300   J = JG1,JG2
C
      JCG = JCG1 + (J-JG1)
C
      DO 300   I = 1,MP
C
      IP1 = I + 1
      XA  = XS(I,J)
      YA  = YS(I,J)
      ZA  = ZS(I,J)
      XB  = XS(IP1,J)
      YB  = YS(IP1,J)
      ZB  = ZS(IP1,J)
```

```
C
      CALL VLINE(AIN,AIT)
C
      KPM = (I-1)*NP + JCG
      KPP = (I-1)*NP + (JCG-1)
      IF(J.LT.JG2) DSN(KPM) = DSN(KPM) - AIN
      IF(J.GT.JG1) DSN(KPP) = DSN(KPP) + AIN
      IF(J.LT.JG2) DST(KPM) = DST(KPM) - AIT
      IF(J.GT.JG1) DST(KPP) = DST(KPP) + AIT
C
  300 CONTINUE
C
C     WRITE ROW OF COEFFICIENTS TO DISK
C
      IRECN = NPANEL*(IBLD-1) + LP
      IRECT = (NPANEL*NBLDS) + IRECN
      WRITE(IOU10, REC = IRECN) (DSN(J), J = 1,NPANEL)
      WRITE(IOU10, REC = IRECT) (DST(J), J = 1,NPANEL)
C
  400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWTO2(AICDW)
C
C     CALCULATES THE INFLUENCE COEFFICIENTS DUE TO THE TRAILING WAKE
C
C     ******************************************************************
C
      SUBROUTINE AICDW(IBLD)
C
      DIMENSION DWN(20),DWT(20)
C
      COMMON/ CONSTS / PI,      RPD
      COMMON/ GVINT  / XA,      YA,      ZA,      XB,      YB,      ZB,
     1                 XC,      YC,      ZC,      AN1C,    AN2C,    AN3C,
     2                 AT1C,    AT2C,    AT3C
      COMMON/ PNLCRD / XS(6,24),        YS(6,24),        ZS(6,24),
     1                 XCOL(5,20),      YCOL(5,20),      ZCOL(5,20)
      COMMON/ PNLNDX / MP,              NP,              NPANEL,
     1                 MP1,             NP1,             NPNLSQ,
     2                 NSGMNT,          JGLIM(4,2),      JCGLIM(4,2),
     3                 NPS(4),          NBLDS
      COMMON/ PNLVEC / AN1(5,20),       AN2(5,20),       AN3(5,20),
     1                 AT1(5,20),       AT2(5,20),       AT3(5,20),
     2                 AQC1(20),        AQC2(20),        AQC3(20)
      COMMON/ UNITS  / IOU1,    IOU5,    IOU6,    IOU7,    IOU8,
     1                 IOU10,   IOU11,   IOU12,   IOU13
      COMMON/ WAKCRD / XW(133,24),      YW(133,24),      ZW(133,24)
      COMMON/ WAKNDX / REVS,            NWIPR,           NWI,
     1                 KMAX(24),        IFWAK,           JS(3,2),
     2                 PSIMR,           PSIMT,           IROLL
C
C     ******************************************************************
C
C     READ SOURCE BLADE WAKE COORDINATES FROM DISK
C
      DO 50   J = 1,NP1
      IREC = (IBLD-1)*NP1+J
      READ(IOU13, REC = IREC) (XW(K,J),YW(K,J),ZW(K,J), K = 1,KMAX(J))
   50 CONTINUE
C
C     SELECT COLLOCATION POINT
C
      DO 500   M = 1,MP
      DO 500   N = 1,NP
C
      LP = (M-1)*NP+N
      XC = XCOL(M,N)
      YC = YCOL(M,N)
      ZC = ZCOL(M,N)
      AN1C = AN1(M,N)
      AN2C = AN2(M,N)
```

```
      AN3C = AN3(M,N)
      AT1C = AT1(M,N)
      AT2C = AT2(M,N)
      AT3C = AT3(M,N)
C
C     INITIALIZE COEFFICIENT MATRICES' ROWS
C
      DO 100   J = 1,NP
      DWN(J) = 0.0
      DWT(J) = 0.0
  100 CONTINUE
C
C     SWEEP SPANWISE ALONG WAKE LEADING EDGE
C
      DO 200   JSEG = 1,NSGMNT
C
      JG1  = JGLIM(JSEG,1)
      JG2  = JGLIM(JSEG,2)
      JCG1 = JCGLIM(JSEG,1)
      JCG2 = JCGLIM(JSEG,2)
C
      DO 200   J = JG1,JG2-1
C
      JCG = JCG1+(J-JG1)
C
C     ASSIGN END POINT COORDINATES TO VORTEX SEGMENT
C
      XA  = XW(1,J)
      YA  = YW(1,J)
      ZA  = ZW(1,J)
      JP1 = J+1
      XB  = XW(1,JP1)
      YB  = YW(1,JP1)
      ZB  = ZW(1,JP1)
C
      CALL VLINE(AIN,AIT)
C
      DWN(JCG) = DWN(JCG)+AIN
      DWT(JCG) = DWT(JCG)+AIT
C
  200 CONTINUE
C
C     SWEEP TRAILERS AND CALCULATE CONTRIBUTION OF EACH
C
      DO 400   JSEG = 1,NSGMNT
C
      JG1  = JGLIM(JSEG,1)
      JG2  = JGLIM(JSEG,2)
      JCG1 = JCGLIM(JSEG,1)
      JCG2 = JCGLIM(JSEG,2)
C
      DO 400   J = JG1,JG2
```

```
C
      JCG = JCG1+(J-JG1)
      K1 = 1
      K2 = KMAX(J)-1
      IFWSKP = 0
      ISTORE = 0
C
C     CHECK IF WITHIN INBOARD OR OUTBOARD SHEET
C
      IF(JS(1,1).LE.J .AND. J.LE.JS(1,2)) GO TO 301
      IF(JS(3,1).LE.J .AND. J.LE.JS(3,2)) GO TO 301
      GO TO 300
C
  301 CONTINUE
      IFWSKP = 1
C
C     SET AZIMUTH LIMITS FOR FIRST PASS ALONG ROLLED-UP VORTEX
C
      IF(J.EQ.1)   K2 = KMAX(2)-1
      IF(J.EQ.NP1) K2 = KMAX(NP1-1)-1
C
C     SUM INFLUENCES OF ELEMENTS ALONG TRAILER
C
  300 CONTINUE
C
      AINS = 0.0
      AITS = 0.0
C
      DO 310   K = K1,K2
C
      XA  = XW(K,J)
      YA  = YW(K,J)
      ZA  = ZW(K,J)
      KP1 = K+1
      XB  = XW(KP1,J)
      YB  = YW(KP1,J)
      ZB  = ZW(KP1,J)
C
      CALL VLINE(AIN,AIT)
C
      AINS = AINS+AIN
      AITS = AITS+AIT
C
  310 CONTINUE
C
      IF(IFWAK.EQ.0)  GO TO 320
      IF(IFWSKP.EQ.1) GO TO 320
C
C     CALCULATE FAR WAKE CONTRIBUTION FROM CYLINDER REPRESENTATION
C
      XA = XB
      YA = YB
```

```
      ZA = ZB
C
      CALL VCYLN(AIN,AIT)
C
      AINS = AINS+AIN
      AITS = AITS+AIT
C
  320 CONTINUE
C
      IF(ISTORE.EQ.1) GO TO 350
C
C     STORE SHEET TRAILER AIC INTO COEFFICIENT ARRAY BUFFERS
C
      JP = JCG-1
      JM = JCG
      IF(J.LT.JG2)  DWN(JM) = DWN(JM)-AINS
      IF(J.GT.JG1)  DWN(JP) = DWN(JP)+AINS
      IF(J.LT.JG2)  DWT(JM) = DWT(JM)-AITS
      IF(J.GT.JG1)  DWT(JP) = DWT(JP)+AITS
C
      IF(J.NE.1 .AND. J.NE.NP1) GO TO 400
C
C     RESET FOR SECOND PASS ALONG ROLLED-UP ROOT OR TIP VORTEX
C
C     CHECK IF ROOT VORTEX IS ROLLED-UP
C
      IF(J.EQ.1 .AND. JS(1,1).EQ.0) GO TO 400
C
C     CHECK IF TIP VORTEX IS ROLLED-UP
C
      IF(J.EQ.NP1 .AND. JS(3,1).EQ.0) GO TO 400
C
C     DETERMINE AIC STORAGE INDEX
C
      IF(J.EQ.1)   JROLL = JS(1,2)
      IF(J.EQ.NP1) JROLL = JS(3,1)
C
C     FIND SEGMENT OF ROLLED UP VORTEX
C
      DO 330 JSEG2=1,NSGMNT
      IF(JGLIM(JSEG2,1).LE.JROLL .AND.
     1   JROLL.LE.JGLIM(JSEG2,2)) GO TO 340
  330 CONTINUE
C
  340 CONTINUE
C
      JCG = JCGLIM(JSEG2,1)+(JROLL-JGLIM(JSEG2,1))
      IF(J.EQ.NP1) JCG = JCG-1
C
      K1 = K2+1
      K2 = KMAX(J)-1
      IFWSKP = 0
```

```
      ISTORE = 1
C
      GO TO 300
C
  350 CONTINUE
C
C     STORE ROLLED-UP VORTEX AIC INTO COEFFICIENT ARRAY BUFFER
C
      IF(J.EQ.1)     DWN(JCG) = DWN(JCG)-AINS
      IF(J.EQ.NP1)   DWN(JCG) = DWN(JCG)+AINS
      IF(J.EQ.1)     DWT(JCG) = DWT(JCG)-AITS
      IF(J.EQ.NP1)   DWT(JCG) = DWT(JCG)+AITS
C
  400 CONTINUE
C
C     WRITE COEFFICIENT MATRICES' ROWS TO DISK
C
      IRECN = NPANEL*(IBLD-1)+LP
      IRECT = NPANEL*NBLDS+IRECN
      WRITE(IOU11,REC=IRECN) (DWN(J), J = 1,NP)
      WRITE(IOU11,REC=IRECT) (DWT(J), J = 1,NP)
C
  500 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT02(AICDSW)
C
C     ASSEMBLES THE AERODYNAMIC INFLUENCE COEFFICIENTS OF THE LIFTING
C     SURFACE AND TRAILED WAKE BY COMBINING THROUGH THE KUTTA CONDITION.
C     THE SELF-INDUCED COMPONENT OF THE AIC'S IS THEN MODIFIED FOR
C     NON-POTENTIAL LIFT CURVE SLOPE.
C
C     ******************************************************************
C
      SUBROUTINE AICDSW(ITBLD)
C
      DIMENSION DSWN(100),DSWT(100),DSN(100),DST(100),DWN(20),DWT(20)
C
      COMMON/ ALSLP  / ALS(20),        ALSNEW(20),     ALSOLD(20)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CDLS   / DLS(25)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     SWEEP TARGET PANELS (SOLUTION POINTS)
C
      DO 600   LP = 1,NPANEL
C
C     INITIALIZE COEFFICIENT MATRICES' ROWS
C
      DO 100   KP = 1,NPANEL
      DSWN(KP) = 0.0
      DSWT(KP) = 0.0
  100 CONTINUE
C
C     COMBINE INFLUENCES OF ALL BLADES AND WAKES
C
      DO 400   ISBLD = 1,NBLDS
C
```

```
C     READ ROWS LP OF COEFFICIENT MATRICES FROM DISK                    00000530
C                                                                       00000540
      IRECN = NPANEL*(ISBLD-1)+LP                                       00000550
      IRECT = NPANEL*NBLDS+IRECN                                        00000560
C                                                                       00000570
C     SURFACE                                                           00000580
C                                                                       00000590
      READ(IOU10, REC = IRECN) (DSN(J), J = 1,NPANEL)                   00000600
      READ(IOU10, REC = IRECT) (DST(J), J = 1,NPANEL)                   00000610
C                                                                       00000620
C     WAKE                                                              00000630
C                                                                       00000640
      READ(IOU11, REC = IRECN) (DWN(J), J = 1,NP)                       00000650
      READ(IOU11, REC = IRECT) (DWT(J), J = 1,NP)                       00000660
C                                                                       00000670
C     ADD SURFACE AND WAKE COEFFICIENTS                                 00000680
C                                                                       00000690
      KP1 = 1                                                           00000700
      KP2 = (MP-2)*NP+NP                                                00000710
C                                                                       00000720
      DO 200   KP = KP1,KP2                                             00000730
      CIRRAT = CIRCR(KP)                                                00000740
      IF(ISBLD.EQ.ITBLD) CIRRAT = 1.0                                   00000750
      DSWN(KP) = DSWN(KP) + (DSN(KP) * CIRRAT)                          00000760
      DSWT(KP) = DSWT(KP) + (DST(KP) * CIRRAT)                          00000770
  200 CONTINUE                                                          00000780
C                                                                       00000790
C     WAKE IS COMBINED THROUGH THE KUTTA CONDITION, EXPRESSED BY THE    00000800
C     RATIO TECF OF WAKE CIRCULATION TO TRAILING EDGE PANEL CIRCULATION 00000810
C                                                                       00000820
      KP1 = KP2+1                                                       00000830
      KP2 = NPANEL                                                      00000840
C                                                                       00000850
      DO 300   KP = KP1,KP2                                             00000860
      JW = KP-KP1+1                                                     00000870
      CIRRAT = CIRCR(KP)                                                00000880
      IF(ISBLD.EQ.ITBLD) CIRRAT = 1.0                                   00000890
      DSWN(KP) = DSWN(KP) + ((DSN(KP) + TECF * DWN(JW)) * CIRRAT)       00000900
      DSWT(KP) = DSWT(KP) + ((DST(KP) + TECF * DWT(JW)) * CIRRAT)       00000910
  300 CONTINUE                                                          00000920
C                                                                       00000930
  400 CONTINUE                                                          00000940
C                                                                       00000950
C     MODIFY FOR NON-POTENTIAL SECTIONAL LIFT CURVE SLOPE               00000960
C                                                                       00000970
      LPJ = LP-((LP-1)/NP)*NP                                           00000980
      LPI = (LP-1)/NP+1                                                 00000990
C                                                                       00001000
      DO 500   I = 1,MP                                                 00001010
      ISTRLS = (I-1)*MP+LPI                                             00001020
      KP = (I-1)*NP+LPJ                                                 00001030
      DSWN(KP) = DSWN(KP)+(2.0*PI/ALS(LPJ)-1.0)*DLS(ISTRLS)/CHDCGL(LPJ) 00001040
```

```
  500 CONTINUE
C
C     WRITE COMBINED COEFFICIENT MATRIX ROW LP TO DISK
C
      IRECN = LP
      IRECT = NPANEL+LP
      WRITE(IOU12, REC = IRECN) (DSWN(KP), KP = 1,NPANEL)
      WRITE(IOU12, REC = IRECT) (DSWT(KP), KP = 1,NPANEL)
C
  600 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWTO2(GSITER)
C
C     GAUSS-SEIDEL ITERATIVE SOLUTION OF LINEAR SIMULTANEOUS EQUATIONS
C     OF THE FORM {A}[X] = [B]
C
C     ******************************************************************
C
      SUBROUTINE GSITER(N)
C
      DIMENSION A(100),X(100),B(100)
C
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
      COMMON/ NMWSH  / NRMWSH(100)
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
      EQUIVALENCE (X,CIRC), (B,NRMWSH)
C
C     ******************************************************************
C
C     NORMALIZE SOLUTION VECTOR
C
      SCLFAC = OMEGAR*RADIUS
      DO 50   I = 1,N
      X(I) = X(I)/SCLFAC
   50 CONTINUE
C
C     SET MAXIMUM TOLERANCE
C
      TOLMAX = 1.0E-05
      IF(TOL1.GT.0.0) TOLMAX = TOL1
C
C     INITIALIZE CONVERGENCE CHECK
C
      SUMX = 0.0
C
C     ALLOW ITCYLM CYLES TO CONVERGE
C
      ITCYLM = ICYL1
      IF(ITCYLM.EQ.0) ITCYLM = 100
C
      DO 100   ITCYL = 1,ITCYLM
C
      SUMXP = 0.0
C
```

```
      DO 200   I = 1,N
C
C     READ ROW I OF COEFFICIENT MATRIX INTO A
C
      READ(IOU12, REC = I) (A(J), J = 1,N)
C
C     CALCULATE NEW APPROXIMATION TO X(I)
C
      AX = B(I)+A(I)*X(I)
C
      DO 300   J = 1,N
      AX = AX-A(J)*X(J)
  300 CONTINUE
C
      X(I) = 0.5*(AX/A(I) - X(I)) + X(I)
C
C     SUM X VECTOR FOR CONVERGENCE CHECK
C
      SUMXP = SUMXP+ABS(X(I))
C
  200 CONTINUE
C
      TOL = ABS((SUMXP-SUMX)/SUMXP)
      IF(TOL.LE.TOLMAX) GO TO 500
C
      SUMX = SUMXP
C
  100 CONTINUE
C
C     SOLUTION VECTOR NOT CONVERGED WITHIN ITMAX ITERATIONS
C
      WRITE(IOU6,10) ITCYLM,TOL,TOLMAX
   10 FORMAT('1',9X,'*** ERROR *** SUBROUTINE GSITER',/
     1       ' ',9X,'CONVERGENCE NOT OBTAINED AT ITERATION ',I3,/
     2       ' ',9X,'CURRENT TOLERANCE   =',1PE12.4,/
     3       ' ',9X,'ALLOWABLE TOLERANCE =',1PE12.4)
C
      STOP ' '
C
  500 CONTINUE
C
C     RESCALE SOLUTION VECTOR
C
      DO 700   I = 1,N
      X(I) = X(I)*SCLFAC
  700 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWTO2(VLINE)
C
C     APPLIES THE BIOT-SAVART LAW TO CALCULATE THE VELOCITY INDUCED BY
C     A FINITE LENGTH LINE VORTEX OF UNIT STRENGTH
C
C     ******************************************************************
C
      SUBROUTINE VLINE(AIN,AIT)
C
      COMMON/ CONSTS / PI,     RPD
      COMMON/ GVINT  / XA,     YA,     ZA,     XB,     YB,     ZB,
     1                 XC,     YC,     ZC,     AN1C,   AN2C,   AN3C,
     2                 AT1C,   AT2C,   AT3C
C
C     ******************************************************************
C
      A  = (XA-XC)**2 + (YA-YC)**2 + (ZA-ZC)**2
      B  = 2.0*( (XA-XB)*(XC-XA) + (YA-YB)*(YC-YA) + (ZA-ZB)*(ZC-ZA) )
      C  = (XA-XB)**2 + (YA-YB)**2 + (ZA-ZB)**2
C
      C1 = (YC-YB)*ZA + (YA-YC)*ZB + (YB-YA)*ZC
      C2 = (ZC-ZB)*XA + (ZA-ZC)*XB + (ZB-ZA)*XC
      C3 = (XC-XB)*YA + (XA-XC)*YB + (XB-XA)*YC
C
      Q  = 4.0*A*C - B**2
C
C     POINTS ARE COLINEAR IF Q=0, THEN QB=0
C
      QB = 0.0
      IF(ABS(Q).GT.(1.0E-10)) QB = 2.0/Q*((2.0*C+B)/SQRT(A+B+C)
     1                            -B/SQRT(A))/4.0/PI
C
      Q1 = C1*QB
      Q2 = C2*QB
      Q3 = C3*QB
C
      AIN = AN1C*Q1 + AN2C*Q2 + AN3C*Q3
      AIT = AT1C*Q1 + AT2C*Q2 + AT3C*Q3
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWTO2(VCYLN)
C
C     USES DIRECT NUMERICAL INTEGRATION TO CALCULATE THE VELOCITY
C     INDUCED BY A SEMI-INFINITE VORTEX CYLINDER OF UNIT STRENGTH
C
C     ******************************************************************
C
      SUBROUTINE VCYLN(AIN,AIT)
C
      COMMON/ CONSTS / PI,     RPD
      COMMON/ GVINT  / XA,     YA,     ZA,     XB,     YB,     ZB,
     1                 XC,     YC,     ZC,     AN1C,   AN2C,   AN3C,
     2                 AT1C,   AT2C,   AT3C
      COMMON/ WRATES / AKZW,   AKZI,   AKZ,    AKR,    A,      AKB,
     1                 CTRW,   AKZO,   ALAMO,  AKRO,   VAXB,   VNUB
C
C     ******************************************************************
C
      RC  = SQRT(XC**2+YC**2)
      RA  = SQRT(XA**2+YA**2)
      ZAC = ZA-ZC
      ZAC = ZAC*SIGN(1.0,AKZW)
      AKZ = AKZW+AKZI
C
C     SET STEP SIZE PROPORTIONAL TO Z
C
      STEPS = 10.0
      ABSZ  = ABS(ZAC)
      IF(ABSZ.LT.1.0) STEPS = -90.0*ABSZ+100.0
      ISTEPS = IFIX(STEPS)
      STEPS  = FLOAT(ISTEPS)
      DPHI   = 2.0*PI/STEPS
C
C     INITIALIZE NUMERICAL INTEGRATION
C
      PHI = DPHI/2.0
      VZ  = 0.0
      VR  = 0.0
C
      DO 100   I = 1,ISTEPS
C
      T1 = RA-RC*COS(PHI)
      T2 = RC**2+RA**2-2.0*RA*RC*COS(PHI)
      T3 = SQRT(RC**2+RA**2+ZAC**2-2.0*RA*RC*COS(PHI))
C
C     AXIAL VELOCITY INDUCED BY SEMI-INFINITE VORTEX CYLINDER
C
      VZ = VZ+RA*T1/T2*(1.0-ZAC/T3)
C
C     RADIAL VELOCITY INDUCED BY SEMI-INFINITE VORTEX CYLINDER
```

```
C
      VR = VR+RA*COS(PHI)/T3
C
      PHI = PHI+DPHI
C
  100 CONTINUE
C
      T4 = -DPHI/8.0/PI**2/AKZ
      T4 = T4*SIGN(1.0,AKZ)
      VZ = VZ*T4
      VR = VR*T4
C
C     RESOLVE FROM CYLINDRICAL TO CARTESIAN
C
      VX = VR*XC/RC
      VY = VR*YC/RC
C
C     NORMAL AND TANGENTIAL INFLUENCE COEFFICIENT COMPONENTS
C
      AIN = AN1C*VX + AN2C*VY + AN3C*VZ
      AIT = AT1C*VX + AT2C*Q2 + AT3C*VZ
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT03(GRID)
C
C     GENERATES NONDIMENSIONAL GRID COORDINATES FOR PANEL AND
C     COLLOCATION POINT LAYOUT.
C
C     CHORDWISE GRID IS A HALF-COSINE DISTRIBUTION BIASED TOWARD THE
C     LEADING EDGE.
C
C     SPANWISE GRID IS OPTIONED ON 'NGRID' FOR EACH SEGMENT:
C         NGRID=1     EQUAL SPACING
C         NGRID=2     FULL-COSINE
C         NGRID=3     HALF-COSINE BIASED INBOARD
C         NGRID=4     HALF-COSINE BIASED OUTBOARD
C
C     ******************************************************************
C
      SUBROUTINE GRID
C
      COMMON/ CGRID  / NGRID(4),       XG(6),  YG(24), XCG(5)
      COMMON/ CONFIG / XOCREF(4),      YSEG(4,2),      JCHORD(4,2),
     1                 YCHORD(24),     CHORD(24),      JTWIST(4,2),
     2                 YTWIST(24),     TWIST(24),      JXAOFF(4,2),
     3                 YXAOFF(24),     XAOFF(24),      JAFT(4,2),
     4                 YAFT(24,2),     IDAF(24),       XTRAN(4),
     5                 YTRAN(4),       ZTRAN(4),       XROT(4),
     6                 YROT(4),        ZROT(4)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
C
C     ******************************************************************
C
      MP1 = MP+1
C
C     CHORDWISE GRID
C
      DPHI = PI/2.0/MP
C
      DO 100   I = 1,MP1
      PHI = (I-1)*DPHI
      XG(I) = 1.0-COS(PHI)
  100 CONTINUE
C
C     SPANWISE GRID FOR EACH SEGMENT
C
      JG2  = 0
      JCG2 = 0
C
```

```
      DO 1000   JSEG = 1,NSGMNT
C
      JG1 = JG2+1
      JG2 = JG1+NPS(JSEG)
      JGLIM(JSEG,1) = JG1
      JGLIM(JSEG,2) = JG2
      JCG1 = JCG2+1
      JCG2 = JCG1+NPS(JSEG)-1
      JCGLIM(JSEG,1) = JCG1
      JCGLIM(JSEG,2) = JCG2
      DYSEG = YSEG(JSEG,2)-YSEG(JSEG,1)
      YG1   = YSEG(JSEG,1)
C
      GO TO (200,300,400,500), NGRID(JSEG)
C
C     EQUAL SPACING
C
  200 CONTINUE
      DELY = DYSEG/NPS(JSEG)
C
      DO 250   J = JG1,JG2
      YG(J) = DELY*(J-JG1)+YG1
  250 CONTINUE
C
      GO TO 1000
C
C     FULL-COSINE SPACING
C
  300 CONTINUE
      DPHI = PI/NPS(JSEG)
C
      DO 350   J = JG1,JG2
      PHI = PI-(J-JG1)*DPHI
      YG(J) = (1.0+COS(PHI))/2.0*DYSEG+YG1
  350 CONTINUE
C
      GO TO 1000
C
C     HALF-COSINE BIASED INBOARD
C
  400 CONTINUE
      DPHI = PI/2.0/NPS(JSEG)
C
      DO 450   J = JG1,JG2
      PHI = PI-(J-JG1)*DPHI
      YG(J) = (1.0+COS(PHI))*DYSEG+YG1
  450 CONTINUE
C
      GO TO 1000
C
C     HALF-COSINE BIASED OUTBOARD
C
```

```
  500 CONTINUE
      DPHI = PI/2.0/NPS(JSEG)
      DO 550   J = JG1,JG2
C
      PHI = PI/2.0-(J-JG1)*DPHI
      YG(J) = COS(PHI)*DYSEG+YG1
  550 CONTINUE
C
 1000 CONTINUE
C
C     PANEL INDEX PARAMETERS
C
      NP  = JCGLIM(NSGMNT,2)
      NP1 = JGLIM(NSGMNT,2)
      NPANEL = MP*NP
      NPNLSQ = NPANEL**2
C
C     COLLOCATION POINT GRID
C
      CALL COLCRD
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT03(COLCRD)
C
C     CALCULATES THE NONDIMENSIONAL CHORDWISE LOCATION OF COLLOCATION
C     POINTS BASED ON TWO-DIMENSIONAL MEAN CIRCULATION
C
C     ******************************************************************
C
      SUBROUTINE COLCRD
C
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CGRID  / NGRID(4),       XG(6),  YG(24), XCG(5)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     MEAN CIRCULATION OVER EACH PANEL
C
      DO 500   I = 1,MP
C
      PHI1 = ACOS(1.0-2.0*XG(I))
      PHI2 = ACOS(1.0-2.0*XG(I+1))
C
      AI1   = SIN(PHI1)-PHI1*COS(PHI1)+PHI1/2.0
     1        -SIN(PHI1)*COS(PHI1)/2.0
      AI2   = SIN(PHI2)-PHI2*COS(PHI2)+PHI2/2.0
     1        -SIN(PHI2)*COS(PHI2)/2.0
      AKBAR = 1.0/2.0/(XG(I+1)-XG(I))*(AI2-AI1)
C
      DO 300   ITER = 1,20
      PHIC = (PHI1+PHI2)/2.0
      FUNCT = PHIC+SIN(PHIC)-AKBAR
      IF(ABS(FUNCT).LE.1.0E-05) GO TO 400
      IF(FUNCT) 100,400,200
  100 PHI1 = PHIC
      GO TO 300
  200 PHI2 = PHIC
  300 CONTINUE
C
      WRITE(IOU6,10) MP,I
   10 FORMAT('0',9X,'XKM ITERATIONS EXCEEDED, MP=',I2,2X,',I=',I2)
      IEND = 1
      GO TO 99
```

```
C
  400 CONTINUE
      XKM(I) = (1.0-COS(PHIC))/2.0
      AKM(I) = AKBAR/PI
C
  500 CONTINUE
C
      AKM(MP1) = 1.0
C
C     CHORDWISE VELOCITY ZERO OF EACH PANEL
C
      DO 1100   I = 1,MP
C
      X1 = XG(I)
      X2 = XG(I+1)
C
      DO 900   ITER = 1,40
      XVN = (X1+X2)/2.0
      VN  = AKM(1)/(XVN-XG(1))
C
      DO 600   J = 1,MP
      VN = VN+(AKM(J+1)-AKM(J))/(XVN-XG(J+1))
  600 CONTINUE
C
      FUNCT = VN-2.0
      IF(ABS(FUNCT).LE.1.0E-05) GO TO 1000
      IF(FUNCT) 700,1000,800
  700 X2 = XVN
      GO TO 900
  800 X1 = XVN
C
  900 CONTINUE
      WRITE(IOU6,12) MP,I
   12 FORMAT('0',9X,'XCG ITERATIONS EXCEEDED, MP=',I2,2X,', I=',I2)
      IEND = 1
      GO TO 99
C
 1000 CONTINUE
      XCG(I) = XVN
C
 1100 CONTINUE
C
      PHIM = ACOS(1.0-2.0*XKM(MP))
      TECF = PI/(PHIM+SIN(PHIM))
C
   99 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT03(RFSURF)
C
C     GENERATES THE PANEL CORNER AND COLLOCATION POINT CARTESIAN
C     COORDINATES OF THE LIFTING SURFACE AT REFERENCE POSITION
C
C     ******************************************************************
C
      SUBROUTINE RFSURF
C
      COMMON/ AFTABL / NAFT,   IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),
     2                 TWSTCG(20)
      COMMON/ CGRID  / NGRID(4),       XG(6),  YG(24), XCG(5)
      COMMON/ CONFIG / XOCREF(4),      YSEG(4,2),      JCHORD(4,2),
     1                 YCHORD(24),     CHORD(24),      JTWIST(4,2),
     2                 YTWIST(24),     TWIST(24),      JXAOFF(4,2),
     3                 YXAOFF(24),     XAOFF(24),      JAFT(4,2),
     4                 YAFT(24,2),     IDAF(24),       XTRAN(4),
     5                 YTRAN(4),       ZTRAN(4),       XROT(4),
     6                 YROT(4),        ZROT(4)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ PNLVEC / AN1(5,20),      AN2(5,20),      AN3(5,20),
     1                 AT1(5,20),      AT2(5,20),      AT3(5,20),
     2                 AQC1(20),       AQC2(20),       AQC3(20)
      COMMON/ REFSUR / XSR(6,24),      YSR(6,24),      ZSR(6,24),
     1                 XCOLR(5,20),    YCOLR(5,20),    ZCOLR(5,20),
     2                 AN1R(5,20),     AN2R(5,20),     AN3R(5,20),
     3                 AT1R(5,20),     AT2R(5,20),     AT3R(5,20)
```

```
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     LOOP THROUGH SEGMENTS
C
      DO 400   JSEG = 1,NSGMNT
C
      JG1  = JGLIM(JSEG,1)
      JG2  = JGLIM(JSEG,2)
      JCG1 = JCGLIM(JSEG,1)
      JCG2 = JCGLIM(JSEG,2)
C
C     LOOP 100 GENERATES PANEL COORDINATES
C
      DO 100   JG = JG1,JG2
C
      CALL LINTRP(YG(JG),CHDGL,JCHORD(JSEG,1),JCHORD(JSEG,2),
     1            YCHORD,CHORD)
      CALL LINTRP(YG(JG),XGOFF,JXAOFF(JSEG,1),JXAOFF(JSEG,2),
     1            YXAOFF,XAOFF)
C
      XLE = XGOFF-XOCREF(JSEG)*CHDGL
C
      CALL AFTLOC(YG(JG),JSEG,JTAB1,JTAB2,AFWF,IEND)
C
      RG = YG(JG)
      ZG = 0.0
      CALL ROTATE(RG,ZG,XROT(JSEG))
      RG = RG + YTRAN(JSEG)
      ZG = ZG + ZTRAN(JSEG)
      IF (ISTDY.EQ.1) BETAO = AB(1,1)
      IF (ISTDY.EQ.1) CALL ROTATE(RG,ZG,BETAO)
      VELGL = SQRT(VWO**2+(OMEGAR*RG)**2)
      REGL = RHO*VELGL*CHDGL*RADIUS/AMU
C
      CALL ZROLFT(AZLL,REGL,JTAB1,JTAB2,AFWF)
C
      CALL LINTRP(YG(JG),TWSTG,JTWIST(JSEG,1),JTWIST(JSEG,2),
     1            YTWIST,TWIST)
C
      ALPT = TWSTG-AZLL
C
      DO 100   I = 1,MP1
C
      XSR(I,JG) = XLE+XG(I)*CHDGL
      YSR(I,JG) = YG(JG)
      ZSR(I,JG) = 0.0
C
C     ROTATE TO ZERO LIFT LINE PLUS TWIST INCREMENT
C
```

```
      CALL ROTATE(ZSR(I,JG),XSR(I,JG),ALPT)
C
C     APPLY SEGMENT INPUT ROTATIONS
C
      CALL ROTATE(ZSR(I,JG),XSR(I,JG),YROT(JSEG))
      CALL ROTATE(YSR(I,JG),ZSR(I,JG),XROT(JSEG))
      CALL ROTATE(XSR(I,JG),YSR(I,JG),ZROT(JSEG))
C
C     APPLY SEGMENT INPUT TRANSLATIONS
C
      XSR(I,JG) = XSR(I,JG)+XTRAN(JSEG)
      YSR(I,JG) = YSR(I,JG)+YTRAN(JSEG)
      ZSR(I,JG) = ZSR(I,JG)+ZTRAN(JSEG)
C
  100 CONTINUE
C
C     LOOP 300 GENERATES COLLOCATION POINT COORDINATES,
C     PANEL ORIENTATION VECTORS, AND SOME SWEEP PARAMETERS
C
      DO 300   JCG = JCG1,JCG2
C
      JG = JG1+JCG-JCG1
C
C     PANEL GRID CENTER COORDINATE, BLADE ELEMENT SPAN
C
      YCG(JCG) = (YG(JG+1)+YG(JG))/2.0
      XBE = 0.0
      YBE = YCG(JCG)
      ZBE = 0.0
      CALL ROTATE(ZBE,XBE,YROT(JSEG))
      CALL ROTATE(YBE,ZBE,XROT(JSEG))
      CALL ROTATE(XBE,YBE,ZROT(JSEG))
      XBE = XBE + XTRAN(JSEG)
      YBE = YBE + YTRAN(JSEG)
      ZBE = ZBE + ZTRAN(JSEG)
      RBE(JCG) = SQRT(XBE**2+YBE**2)
      IF (ISTDY.EQ.1) BETA0 = AB(1,1)
      IF (ISTDY.EQ.1) CALL ROTATE(YBE,ZBE,BETA0)
      DBES(JCG) = YG(JG+1)-YG(JG)
C
C     LEADING AND TRAILING EDGE, AND QUARTER CHORD DIRECTION VECTORS
C     FOR USE IN SWEEP CALCULATIONS
C
      ALE1 = XSR(1,JG+1) - XSR(1,JG)
      ALE2 = YSR(1,JG+1) - YSR(1,JG)
      ALE3 = ZSR(1,JG+1) - ZSR(1,JG)
C
      ATE1 = XSR(MP1,JG+1) - XSR(MP1,JG)
      ATE2 = YSR(MP1,JG+1) - YSR(MP1,JG)
      ATE3 = ZSR(MP1,JG+1) - ZSR(MP1,JG)
C
      AQC1(JCG) = ALE1 + 0.25*(ATE1-ALE1)
```

```
      AQC2(JCG) = ALE2 + 0.25*(ATE2-ALE2)
      AQC3(JCG) = ALE3 + 0.25*(ATE3-ALE3)
C
C     PUT QUARTER CHORD VECTOR BACK INTO BLADE ELEMENT REFERENCE
C
      ANGLE = - XROT(JSEG)
      CALL ROTATE(AQC2(JCG),AQC3(JCG),ANGLE)
C
      CALL LINTRP(YCG(JCG),CHDCGL(JCG),JCHORD(JSEG,1),JCHORD(JSEG,2),
     1            YCHORD,CHORD)
      CALL LINTRP(YCG(JCG),XCGOFF,JXAOFF(JSEG,1),JXAOFF(JSEG,2),
     1            YXAOFF,XAOFF)
C
      XCLE(JCG) = XCGOFF-XOCREF(JSEG)*CHDCGL(JCG)
C
C     GET ZERO LIFT LINE, THICKNESS, AND TWIST
C
      CALL AFTLOC(YCG(JCG),JSEG,JTAB(JCG,1),JTAB(JCG,2),AFTWF(JCG),IEND)
C
      TOC1 = AFTOC(JTAB(JCG,1))
      TOC2 = AFTOC(JTAB(JCG,2))
      TOC(JCG) = TOC2*AFTWF(JCG)+TOC1*(1.0-AFTWF(JCG))
      VELCGL = SQRT(VWO**2+(OMEGAR*RBE(JCG))**2)
      RECGL  = RHO*VELCGL*CHDCGL(JCG)*RADIUS/AMU
C
      CALL ZROLFT(AZLLC(JCG),RECGL,JTAB(JCG,1),JTAB(JCG,2),AFTWF(JCG))
C
      CALL LINTRP(YCG(JCG),TWSTCG(JCG),JTWIST(JSEG,1),JTWIST(JSEG,2),
     1            YTWIST,TWIST)
C
C     ROTATE TO ZERO LIFT LINE PLUS TWIST INCREMENT
C
      ALPT = TWSTCG(JCG)-AZLLC(JCG)
C
      DO 200   I = 1,MP
C
      XCOLR(I,JCG) = XCLE(JCG)+XCG(I)*CHDCGL(JCG)
      YCOLR(I,JCG) = YCG(JCG)
      ZCOLR(I,JCG) = 0.0
C
      CALL ROTATE(ZCOLR(I,JCG),XCOLR(I,JCG),ALPT)
C
C     REFERENCE PANEL DIRECTION VECTORS
C
      AN1R(I,JCG) = 0.0
      AN2R(I,JCG) = 0.0
      AN3R(I,JCG) = 1.0
C
      CALL ROTATE(AN3R(I,JCG),AN1R(I,JCG),ALPT)
C
      AT1R(I,JCG) = 1.0
      AT2R(I,JCG) = 0.0
```

```
      AT3R(I,JCG) = 0.0
C
      CALL ROTATE(AT3R(I,JCG),AT1R(I,JCG),ALPT)
C
C     APPLY SEGMENT INPUT ROTATIONS
C
      CALL ROTATE(ZCOLR(I,JCG),XCOLR(I,JCG),YROT(JSEG))
      CALL ROTATE(YCOLR(I,JCG),ZCOLR(I,JCG),XROT(JSEG))
      CALL ROTATE(XCOLR(I,JCG),YCOLR(I,JCG),ZROT(JSEG))
C
      CALL ROTATE(AN3R(I,JCG),AN1R(I,JCG),YROT(JSEG))
      CALL ROTATE(AN2R(I,JCG),AN3R(I,JCG),XROT(JSEG))
      CALL ROTATE(AN1R(I,JCG),AN2R(I,JCG),ZROT(JSEG))
C
      CALL ROTATE(AT3R(I,JCG),AT1R(I,JCG),YROT(JSEG))
      CALL ROTATE(AT2R(I,JCG),AT3R(I,JCG),XROT(JSEG))
      CALL ROTATE(AT1R(I,JCG),AT2R(I,JCG),ZROT(JSEG))
C
C     APPLY SEGMENT INPUT TRANSLATIONS
C
      XCOLR(I,JCG) = XCOLR(I,JCG)+XTRAN(JSEG)
      YCOLR(I,JCG) = YCOLR(I,JCG)+YTRAN(JSEG)
      ZCOLR(I,JCG) = ZCOLR(I,JCG)+ZTRAN(JSEG)
C
  200 CONTINUE
C
C     COSINE OF LEADING EDGE SWEEP ANGLE
C
      CALL VPANG(ALE1,ALE2,ALE3,AT1R(1,JCG),AT2R(1,JCG),AT3R(1,JCG),
     1           CLAMLE(JCG))
C
  300 CONTINUE
C
  400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT03(AFTLOC)
C
C     ASSIGNS AIRFOIL TABLE STORAGE LOCATION INDICES TO BLADE GRID
C     STATION Y
C
C     ******************************************************************
C
      SUBROUTINE AFTLOC(Y,JSEG,JTAB1,JTAB2,AFTWF,IEND)
C
      COMMON/ AFTABL / NAFT,   IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
      COMMON/ CONFIG / XOCREF(4),      YSEG(4,2),      JCHORD(4,2),
     1                 YCHORD(24),     CHORD(24),      JTWIST(4,2),
     2                 YTWIST(24),     TWIST(24),      JXAOFF(4,2),
     3                 YXAOFF(24),     XAOFF(24),      JAFT(4,2),
     4                 YAFT(24,2),     IDAF(24),       XTRAN(4),
     5                 YTRAN(4),       ZTRAN(4),       XROT(4),
     6                 YROT(4),        ZROT(4)
C
C     ******************************************************************
C
      IEND = 0
C
C     SET SEGMENT PARAMETERS
C
      J1 = JAFT(JSEG,1)
      J2 = JAFT(JSEG,2)
C
C     CHECK THAT Y IS WITHIN RANGE, AND IF NOT SET LOCATION INDICES
C     TO APPROPRIATE BOUNDING TABLES FOR THIS SEGMENT
C
      IDAF1 = IDAF(J1)
      IDAF2 = IDAF(J1)
      IF(Y.LT.YAFT(J1,1)) GO TO 400
      IDAF1 = IDAF(J2)
      IDAF2 = IDAF(J2)
      IF(Y.GT.YAFT(J2,2)) GO TO 400
C
C     TRAP Y WITHIN TABLE RANGE
C
      DO 100   J = J1,J2
      IF(Y.GE.YAFT(J,1).AND.Y.LE.YAFT(J,2)) GO TO 200
      IF(Y.LT.YAFT(J+1,1))                  GO TO 300
  100 CONTINUE
C
  200 CONTINUE
```

```
C
C     Y IS WITHIN AIRFOIL TABLE RANGE
C
      IDAF1 = IDAF(J)
      IDAF2 = IDAF(J)
      AFTWF = 0.50
      IF(J.EQ.J2) GO TO 400
C
C     CHECK FOR Y EQUAL TO AIRFOIL DISCONTINUITY STATION
C
      IF(Y.NE.YAFT(J+1,1)) GO TO 400
      IDAF2 = IDAF(J+1)
      AFTWF = 0.50
      GO TO 400
C
  300 CONTINUE
C
C     Y IS IN TRANSITION REGION BETWEEN AIRFOIL TABLE RANGES
C
      IDAF1 = IDAF(J)
      IDAF2 = IDAF(J+1)
      AFTWF = 0.50
      IF(YAFT(J+1,1)-YAFT(J,2).LT.1.0E-05) GO TO 400
      AFTWF = (Y-YAFT(J,2))/(YAFT(J+1,1)-YAFT(J,2))
C
  400 CONTINUE
C
C     ASSIGN TABLE LOCATION INDEX BY MATCHING IDAF TO IDAFT
C
      JTAB1 = 0
      JTAB2 = 0
C
      DO 500   J = 1,NAFT
      IF(IDAF1.EQ.IDAFT(J)) JTAB1 = J
      IF(IDAF2.EQ.IDAFT(J)) JTAB2 = J
  500 CONTINUE
C
      IF(JTAB1.EQ.0.OR.JTAB2.EQ.0) IEND = 1
C
      RETURN
      END
```

```
C     ******************************************************************
C                                                                      
C     LSWT03(LINTRP)                                                   
C                                                                      
C     PERFORMS A LINEAR INTERPOLATION OF A PIECEWISE LINEAR FUNCTION.  
C     IF TARGET IS OUTSIDE RANGE, THEN EXTRAPOLATION IS USED.          
C                                                                      
C     ******************************************************************
C                                                                      
      SUBROUTINE LINTRP(YINT,FINT,JF1,JF2,YF,F)                        
C                                                                      
      DIMENSION YF(1),F(1)                                             
C                                                                      
C     ******************************************************************
C                                                                      
      DO 100   J = JF1+1,JF2                                           
      IF(YF(J-1).LE.YINT.AND.YINT.LE.YF(J)) GO TO 200                  
  100 CONTINUE                                                         
C                                                                      
      IF(YINT.LT.YF(JF1)) J = JF1+1                                    
      IF(YINT.GT.YF(JF2)) J = JF2                                      
C                                                                      
  200 CONTINUE                                                         
      DFDY = (F(J)-F(J-1))/(YF(J)-YF(J-1))                             
      FINT = DFDY*(YINT-YF(J-1))+F(J-1)                                
C                                                                      
      RETURN                                                           
      END                                                              
```

```
C     ******************************************************************
C
C     LSWT03(ROTATE)
C
C     PERFORMS THE ROTATION THROUGH ANGLE OF (X,Y).  X-AXIS INTO Y-AXIS
C     IS POSITIVE FOR POSITIVE ANGLE.
C
C     ******************************************************************
C
      SUBROUTINE ROTATE(X,Y,ANGLE)
C
C     ******************************************************************
C
      SINANG = SIN(ANGLE)
      COSANG = COS(ANGLE)
C
      XP = X*COSANG-Y*SINANG
      YP = X*SINANG+Y*COSANG
      X = XP
      Y = YP
C
      RETURN
      END
```

```
C     ******************************************************************00000010
C                                                                       00000020
C     LSWT03(PSURF)                                                     00000030
C                                                                       00000040
C     POSITIONS REFERENCE SURFACE TO ACCOUNT FOR BLADE PITCH, AZIMUTH,  00000050
C     AND FLAPPING.  IBLD=1 ACTS UPON BOTH GRID AND COLLOCATION POINT   00000060
C     COORDINATES.  IBLD>1 ACTS ONLY UPON GRID COORDINATES.             00000070
C                                                                       00000080
C     ******************************************************************00000090
C                                                                       00000100
      SUBROUTINE PSURF(ALPHAS,PSI,IBLD)                                 00000110
C                                                                       00000120
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),         00000130
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),      00000140
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),       00000150
     3                 JTAB(20,2),     TOC(20),        RE(20),          00000160
     4                 CLT(20),        CDT(20),        CMT(20),         00000170
     5                 CDS(20)                                          00000180
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),      00000190
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),       00000200
     2                 TWSTCG(20)                                       00000210
      COMMON/ CONFIG / XOCREF(4),      YSEG(4,2),      JCHORD(4,2),     00000220
     1                 YCHORD(24),     CHORD(24),      JTWIST(4,2),     00000230
     2                 YTWIST(24),     TWIST(24),      JXAOFF(4,2),     00000240
     3                 YXAOFF(24),     XAOFF(24),      JAFT(4,2),       00000250
     4                 YAFT(24,2),     IDAF(24),       XTRAN(4),        00000260
     5                 YTRAN(4),       ZTRAN(4),       XROT(4),         00000270
     6                 YROT(4),        ZROT(4)                          00000280
      COMMON/ CONSTS / PI,     RPD                                      00000290
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,               00000300
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,            00000310
     2                 TO,     TP,     TPO,    TPS,    TK,              00000320
     3                 AB(5,2),        AT(5,2)                          00000330
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG                   00000340
      COMMON/ PNLCRD / XS(6,24),       YS(6,24),       ZS(6,24),        00000350
     1                 XCOL(5,20),     YCOL(5,20),     ZCOL(5,20)       00000360
      COMMON/ PNLNDX / MP,             NP,             NPANEL,          00000370
     1                 MP1,            NP1,            NPNLSQ,          00000380
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),     00000390
     3                 NPS(4),         NBLDS                            00000400
      COMMON/ PNLVEC / AN1(5,20),      AN2(5,20),      AN3(5,20),       00000410
     1                 AT1(5,20),      AT2(5,20),      AT3(5,20),       00000420
     2                 AQC1(20),       AQC2(20),       AQC3(20)         00000430
      COMMON/ REFSUR / XSR(6,24),      YSR(6,24),      ZSR(6,24),       00000440
     1                 XCOLR(5,20),    YCOLR(5,20),    ZCOLR(5,20),     00000450
     2                 AN1R(5,20),     AN2R(5,20),     AN3R(5,20),      00000460
     3                 AT1R(5,20),     AT2R(5,20),     AT3R(5,20)       00000470
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,            00000480
     1                 IOU10,  IOU11,  IOU12,  IOU13                    00000490
C                                                                       00000500
C     ******************************************************************00000510
C                                                                       00000520
```

```
      ALPHAP = ALPHAS
      IF(ISTDY.EQ.0) GO TO 130
C
C     FLAPPING ANGULAR DISPLACEMENT
C
      BETA = AB(1,1)
      NBH  = AB(1,2)+0.1
      DO 110   N = 1,NBH
      BETA = BETA+AB(N+1,1)*COS(N*PSI)+AB(N+1,2)*SIN(N*PSI)
  110 CONTINUE
C
C     CYCLIC PITCH
C
      ALPHAC = AT(1,1)
      NTH    = AT(1,2)+0.1
      DO 120   N = 1,NTH
      ALPHAC = ALPHAC+AT(N+1,1)*COS(N*PSI)+AT(N+1,2)*SIN(N*PSI)
  120 CONTINUE
C
C     TOTAL PITCH ANGLE CHANGE FROM REFERENCE BLADE VALUE
C
      ALPHAP = ALPHAP+ALPHAC
C
  130 CONTINUE
C
      DO 100   I = 1,MP1
      DO 100   J = 1,NP1
C
C     LOAD SURFACE PANEL COORDINATES FROM REFERENCE SET
C
      XS(I,J) = XSR(I,J)
      YS(I,J) = YSR(I,J)
      ZS(I,J) = ZSR(I,J)
C
C     ROTATE TO PITCH ABOUT Y-AXIS
C
      CALL ROTATE(ZS(I,J),XS(I,J),ALPHAP)
C
C     ROTATE TO BETA ABOUT X-AXIS
C
      IF(ISTDY.EQ.1) CALL ROTATE(YS(I,J),ZS(I,J),BETA)
C
C     ROTATE TO PSI ABOUT Z-AXIS
C
      CALL ROTATE(XS(I,J),YS(I,J),PSI)
C
  100 CONTINUE
C
      IF(IBLD.GT.1) GO TO 400
C
      DO 300   JSEG = 1, NSGMNT
      JCG1 = JCGLIM(JSEG,1)
```

```
      JCG2 = JCGLIM(JSEG,2)
      DO 300   J = JCG1, JCG2
      DO 200   I = 1, MP
C
C     LOAD PANEL COLLOCATION COORDINATES FROM REFERENCE SET
C
      XCOL(I,J) = XCOLR(I,J)
      YCOL(I,J) = YCOLR(I,J)
      ZCOL(I,J) = ZCOLR(I,J)
C
C     ROTATE TO PITCH ABOUT Y-AXIS
C
      CALL ROTATE(ZCOL(I,J),XCOL(I,J),ALPHAP)
C
C     ROTATE TO BETA ABOUT X-AXIS
C
      IF(ISTDY.EQ.1) CALL ROTATE(YCOL(I,J),ZCOL(I,J),BETA)
C
C     ROTATE TO PSI ABOUT Z-AXIS
C
      CALL ROTATE(XCOL(I,J),YCOL(I,J),PSI)
C
C     LOAD PANEL ORIENTATION VECTORS FROM REFERENCE SET
C
      AN1(I,J) = AN1R(I,J)
      AN2(I,J) = AN2R(I,J)
      AN3(I,J) = AN3R(I,J)
C
      AT1(I,J) = AT1R(I,J)
      AT2(I,J) = AT2R(I,J)
      AT3(I,J) = AT3R(I,J)
C
C     REPEAT PROCEEDURES AS ABOVE
C
      CALL ROTATE(AN3(I,J),AN1(I,J),ALPHAP)
      CALL ROTATE(AT3(I,J),AT1(I,J),ALPHAP)
C
      IF(ISTDY.EQ.1) CALL ROTATE(AN2(I,J),AN3(I,J),BETA)
      IF(ISTDY.EQ.1) CALL ROTATE(AT2(I,J),AT3(I,J),BETA)
C
      CALL ROTATE(AN1(I,J),AN2(I,J),PSI)
      CALL ROTATE(AT1(I,J),AT2(I,J),PSI)

  200 CONTINUE
C
C     ELEMENT GEOMETRIC ANGLE OF ATTACK
C
      ALPHAG(J) = ALPHAP+TWSTCG(J)
C
C     BLADE ELEMENT INCLINATION FROM X-Y PLANE (LOCAL CONING + FLAPPING)
C
      BETAL(J) = XROT(JSEG)
```

```
      IF(ISTDY.EQ.1) BETAL(J) = BETAL(J) + BETA
C
  300 CONTINUE
C
  400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************00000010
C                                                                        00000020
C     LSWT03(VPANG)                                                      00000030
C                                                                        00000040
C     RETURNS THE SINE 'SINTH' OF THE ANGLE BETWEEN THE VECTORS 'A' AND 00000050
C     'B' BASED ON THE DEFINITION OF THE VECTOR PRODUCT                  00000060
C                                                                        00000070
C     ******************************************************************00000080
C                                                                        00000090
      SUBROUTINE VPANG(AX,AY,AZ,BX,BY,BZ,SINTH)                          00000100
C                                                                        00000110
C     ******************************************************************00000120
C                                                                        00000130
      AXBX = AY*BZ - AZ*BY                                               00000140
      AXBY = AZ*BX - AX*BZ                                               00000150
      AXBZ = AX*BY - AY*BX                                               00000160
C                                                                        00000170
      ASQ   = AX**2   + AY**2   + AZ**2                                  00000180
      BSQ   = BX**2   + BY**2   + BZ**2                                  00000190
      AXBSQ = AXBX**2 + AXBY**2 + AXBZ**2                                00000200
C                                                                        00000210
      SINTH = SQRT(AXBSQ/(ASQ*BSQ))                                      00000220
      IF(SINTH.GT.1.0) SINTH = 1.0                                       00000230
C                                                                        00000240
      RETURN                                                             00000250
      END                                                                00000260
```

```
C     ******************************************************************
C
C     LSWT04(PREWAK)
C
C     GENERATES THE DISCRETIZED WAKE VORTEX FILAMENTS OF THE PRESCRIBED
C     WAKE MODEL
C
C     ******************************************************************
C
      SUBROUTINE PREWAK(PSIBIN,IBLD)
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
C
C     ******************************************************************
C
C     PUT PSIB IN 0-->2*PI RANGE
C
      PSIB = 2.0*PI*(PSIBIN/2.0/PI-IFIX(PSIBIN/2.0/PI))
C
C     AZIMUTH INCREMENT PARAMETERS
C
      ANWI = REVS*NWIPR
      NWI  = IFIX(ANWI)
      IF(ANWI-NWI.GT.0.5) NWI = NWI+1
      NWI  = NWI+NWIPR+1
C
      DO 100   J = 1,JGLIM(NSGMNT,2)
      KMAX(J) = NWI
  100 CONTINUE
C
C     GET WAKE PARAMETERS
C
      CALL WAKPRM
      IF(IPRT7.GT.0) CALL TRACE('WAKPRM')
C
      IF(JS(1,1).EQ.0) GO TO 200
C
C     ROOT VORTEX FILAMENT
C
      J1 = JS(1,1)
      J2 = J1
```

```
C
      CALL FILMNT(J1,J2,IBLD,PSIB)
C
  200 CONTINUE
C
      IF(JS(3,1).EQ.0) GO TO 300
C
C     TIP VORTEX FILAMENT
C
      J1 = JS(3,2)
      J2 = J1
C
      CALL FILMNT(J1,J2,IBLD,PSIB)
C
  300 CONTINUE
C
      IF(JS(2,1).EQ.0) GO TO 400
C
C     MID-SHEET
C
      J1 = JS(2,1)
      J2 = JS(2,2)
C
      CALL FILMNT(J1,J2,IBLD,PSIB)
C
  400 CONTINUE
C
      IF(JS(1,1).EQ.0) GO TO 500
C
C     INBOARD SHEET ROLLUP
C
      J1 = JS(1,1)
      J2 = JS(1,2)
      JM = J1
C
      CALL FILROL(JM,J1,J2,IBLD,PSIB)
C
  500 CONTINUE
C
      IF(JS(3,1).EQ.0) GO TO 600
C
C     OUTBOARD SHEET ROLLUP
C
      J1 = JS(3,1)
      J2 = JS(3,2)
      JM = J2
C
      CALL FILROL(JM,J1,J2,IBLD,PSIB)
C
  600 CONTINUE
C
      RETURN
```

```
      END                                                       00001050
```

```
C     ******************************************************************
C
C     LSWT04(FILMNT)
C
C     CALCULATES THE TRAJECTORY OF A FILAMENT USING THE PRESCRIBED WAKE
C     EQUATIONS FOR ROOT AND TIP VORTICES AND MID-SHEET
C
C     ******************************************************************
C
      SUBROUTINE FILMNT(J1,J2,IBLD,PSIB)
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PNLCRD / XS(6,24),       YS(6,24),       ZS(6,24),
     1                 XCOL(5,20),     YCOL(5,20),     ZCOL(5,20)
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WAKCRD / XW(133,24),     YW(133,24),     ZW(133,24)
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
      COMMON/ WRATES / AKZW,   AKZI,   AKZ,    AKR,    A,      AKB,
     1                 CTRW,   AKZO,   ALAMO,  AKRO,   VAXB,   VNUB
C
C     ******************************************************************
C
C     MAXIMUM WAKE RADIUS AT BLADE TRAILING EDGE
C
      RWMAX = SQRT(XS(MP1,NP1)**2+YS(MP1,NP1)**2+ZS(MP1,NP1)**2)
C
      IF(ISTDY.EQ.0) CALL WNDPRO(0,0.0,0.0,0.0,VW1,VW2,VW3,AKZW)
      IF((ISTDY.EQ.0) .AND. (IPRT7.GT.0)) CALL TRACE('WNDPRO')
C
      DO 500   J = J1,J2
C
C     STARTING COORDINATE  OF TRAILER FILAMENT IS BLADE TRAILING EDGE
C
      XFO  = XS(MP1,J)
      YFO  = YS(MP1,J)
      ZFO  = ZS(MP1,J)
      RFO    = SQRT(XFO**2+YFO**2+ZFO**2)
      RRATIO = RFO/RWMAX
```

```
C
      XP = XFO
      YP = YFO
      CALL ROTATE(XP,YP,-PSIB)
      PSIO = ATAN2(-XP,YP)
      IF(PSIO.GT.0.0) PSIO = PSIO-2.0*PI
      PSIO = PSIO+PSIB
C
C     GENERATE FILAMENT COORDINATES
C
      XW(1,J) = XFO
      YW(1,J) = YFO
      ZW(1,J) = ZFO
C
      IF(ISTDY.EQ.0) GO TO 110
C
      BETAB = AB(1,1)
      NBH   = IFIX(AB(1,2)+0.1)
      DO 100   N = 1,NBH
      BETAB = BETAB+AB(N+1,1)*COS(N*PSIB)+AB(N+1,2)*SIN(N*PSIB)
  100 CONTINUE
C
  110 CONTINUE
C
C     SWEEP AZIMUTH GRID
C
      FNWI = FLOAT(NWIPR)
      DO 400 K = 2,KMAX(J)
      FKM1 = FLOAT(K-1)
      IF(K.LE.2*NWIPR+1) PSIW = PSIO-(2.0*PI-(PSIB-PSIO))*
     1                          (FKM1/(2.0*FNWI))**2
      IF(K.GT.2*NWIPR+1) PSIW = PSIB-2.0*PI/FNWI*(FKM1-FNWI)
C
      IF(ISTDY.EQ.0) GO TO 210
      BETAW = AB(1,1)
      PSIOW = PSIW+(PSIB-PSIO)
      DO 200   N = 1,NBH
      BETAW = BETAW+AB(N+1,1)*COS(N*PSIOW)+AB(N+1,2)*SIN(N*PSIOW)
  200 CONTINUE
C
  210 CONTINUE
C
C     RELOCATE STARTING COORDINATES BACK IN AZIMUTH (TIME)
C
      XF = XFO
      YF = YFO
      ZF = ZFO
C
      IF(ISTDY.EQ.1) CALL ROTATE(YF,ZF,BETAW-BETAB)
      CALL ROTATE(XF,YF,PSIW-PSIO)
C
      IF(ISTDY.EQ.0) GO TO 220
```

```
C
C     TRANSFORM INITIAL POINT TO WIND AXIS FOR INTEGRATION
C
      CALL ROTATE(ZF,XF,ALPS2)
      CALL ROTATE(YF,ZF,ALPS1)
C
  220 CONTINUE
C
C     INTEGRATE WITH KS STEPS OVER TIME TO GET PRESENT POSITION
C
      KS = K
      IF(ISTDY.EQ.0) KS = 1
      DPSI = -(PSIW-PSIO)/KS
      PSIF = ATAN2(-XF,YF)
C
      DO 300   I = 1,KS
C
C     GET SETTLING RATE DUE TO WIND PROFILE
C
      IF(ISTDY.EQ.1) CALL WNDPRO(0,XF,YF,ZF,VW1,VW2,VW3,AKZW)
C
C     UPDATE TRAILER COORDINATE FOR THIS STEP

      ZF = ZF+(AKZW+AKZI)*DPSI
      RF = RRATIO*(A+(1-A)*EXP(-AKR*(ZF-ZFO)))
      XF = -RF*SIN(PSIF)
      YF = RF*COS(PSIF)
C
  300 CONTINUE
C
      IF(ISDTY.EQ.0) GO TO 310
C
C     TRANSFORM BACK TO SHAFT AXIS
C
      CALL ROTATE(YF,ZF,-ALPS1)
      CALL ROTATE(ZF,XF,-ALPS2)
      RF = SQRT(XF**2+YF**2)
C
  310 CONTINUE
C
C     SAVE FILAMENT INTO STORAGE ARRAY
C
      XW(K,J) = XF
      YW(K,J) = YF
      ZW(K,J) = ZF
C
  400 CONTINUE
C
C     LOAD FILAMENT COORDINATES INTO DISK FILE
C
      IREC = (IBLD-1)*NP1+J
      WRITE(IOU13, REC = IREC) (XW(K,J),YW(K,J),ZW(K,J), K = 1,KMAX(J))
```

```
C                                                                       00001570
  500 CONTINUE                                                          00001580
C                                                                       00001590
      RETURN                                                            00001600
      END                                                               00001610
```

```
C     ******************************************************************
C
C     LSWT04(FILROL)
C
C     FAIRS INBOARD OR OUTBOARD SHEET FILAMENTS FROM BLADE TO MERGE
C     POINT ON ROOT OR TIP VORTEX TRAJECTORY RESPECTIVELY
C
C     ******************************************************************
C
      SUBROUTINE FILROL(JM,J1,J2,IBLD,PSIB)
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLCRD / XS(6,24),       YS(6,24),       ZS(6,24),
     1                 XCOL(5,20),     YCOL(5,20),     ZCOL(5,20)
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WAKCRD / XW(133,24),     YW(133,24),     ZW(133,24)
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
C
C     ******************************************************************
C
C     READ COORDINATES FROM DISK FILE FOR FILAMENT TO MERGE INTO
C
      IREC = (IBLD-1)*NP1+JM
      READ(IOU13, REC = IREC) (XW(K,JM),YW(K,JM),ZW(K,JM), K=1,KMAX(JM))
C
C     SEARCH TO LOCATE MERGE POINT NEAREST DESIRED AZIMUTH
C
      IF(JM.EQ.J1) PSIMIN = PSIMR+PSIB
      IF(JM.EQ.J2) PSIMIN = PSIMT+PSIB
C
      DO 100   K = 1,KMAX(JM)
C
      XM   = XW(K,JM)
      YM   = YW(K,JM)
      CALL ROTATE(XM,YM,-PSIB)
      PSIM = ATAN2(-XM,YM)
      IF(PSIM.GT.0.0) PSIM = PSIM-2.0*PI
      PSIM = PSIM+PSIB
C
      IF(PSIM.LE.PSIMIN) GO TO 200
C
  100 CONTINUE
C
```

```
  200 CONTINUE
C
      KM = K
      RM = SQRT(XW(KM,JM)**2+YW(KM,JM)**2)
      ZM = ZW(KM,JM)
C
      RMM1 = SQRT(XW(KM-1,JM)**2+YW(KM-1,JM)**2)
      XM   = XW(KM-1,JM)
      YM   = YW(KM-1,JM)
      CALL ROTATE(XM,YM,-PSIB)
      PSIMM1 = ATAN2(-XM,YM)
      IF(PSIMM1.GT.0.0) PSIMM1 = PSIMM1-2.0*PI
      PSIMM1 = PSIMM1+PSIB
C
      RMP1 = SQRT(XW(KM+1,JM)**2+YW(KM+1,JM)**2)
      XM   = XW(KM+1,JM)
      YM   = YW(KM+1,JM)
      CALL ROTATE(XM,YM,-PSIB)
      PSIMP1 = ATAN2(-XM,YM)
      IF(PSIMP1.GT.0.0) PSIMP1 = PSIMP1-2.0*PI
      PSIMP1 = PSIMP1+PSIB
C
      AA = ((RMP1-RMM1)/(PSIMP1-PSIMM1)+(RMM1-RM)/(PSIM-PSIMM1))/
     1     ((PSIMP1**2-PSIMM1**2)/(PSIMP1-PSIMM1)-(PSIM**2-PSIMM1**2)/
     2     (PSIM-PSIMM1))
      BB = (RM-RMM1)/(PSIM-PSIMM1)-(PSIM**2-PSIMM1**2)/(PSIM-PSIMM1)*AA
C
      RMP = 2.0*AA*PSIM+BB
C
C     SWEEP RADIALLY
C
      IF(JM.EQ.J1) J1 = J1+1
      IF(JM.EQ.J2) J2 = J2-1
      DO 300   J = J1,J2
C
C     STARTING COORDINATE  OF TRAILER FILAMENT IS BLADE TRAILING EDGE
C
      XFO  = XS(MP1,J)
      YFO  = YS(MP1,J)
      ZFO  = ZS(MP1,J)
      RFO  = SQRT(XFO**2+YFO**2)
C
      XP = XFO
      YP = YFO
      CALL ROTATE(XP,YP,-PSIB)
      PSIO = ATAN2(-XP,YP)
      IF(PSIO.GT.0.0) PSIO = PSIO-2.0*PI
      PSIO = PSIO+PSIB
C
      XW(1,J) = XFO
      YW(1,J) = YFO
      ZW(1,J) = ZFO
```

```
C
C     FAIRING CONSTANTS
C
      DPSI = PSIM-PSIO
      AA   = (2.0*RFO+RMP*DPSI-2.0*RM)/DPSI**3
      BB   = (RMP-3.0*AA*DPSI**2)/2.0/DPSI
      ZSLP = (ZM-ZFO)/(PSIM-PSIO)
C
C     SET LENGTH INDEX TO KM
C
      KMAX(J) = KM
C
C     SWEEP AND GENERATE AZIMUTH GRID
C
      FKMM1 = FLOAT(KM-1)
C
      DO 400   K = 2,KM
C
      FKM1 = FLOAT(K-1)
      PSIW = PSIO+(PSIM-PSIO)*(FKM1/FKMM1)**2
      DPSI = PSIW-PSIO
C
C     FAIR FROM TRAILING EDGE TO MERGE POINT WITH CUBIC IN AZIMUTH
C     FOR R AND LINEAR IN AZIMUTH FOR Z
C
      RF = AA*DPSI**3+BB*DPSI**2+RFO
      ZF = ZSLP*DPSI+ZFO
      XF = -RF*SIN(PSIW)
      YF = RF*COS(PSIW)
C
C     SAVE COORDINATE INTO STORAGE ARRAY
C
      XW(K,J) = XF
      YW(K,J) = YF
      ZW(K,J) = ZF
C
  400 CONTINUE
C
C     LOAD FILAMENT COORDINATES INTO DISK FILE
C
      IREC = (IBLD-1)*NP1+J
      WRITE(IOU13, REC = IREC) (XW(K,J),YW(K,J),ZW(K,J), K = 1,KMAX(J))
C
  300 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT04(PWAKE)
C
C     POSITIONS THE WAKE COORDINATES FOR EACH BLADE IF ISTDY = 0
C
C     ******************************************************************
C
      SUBROUTINE PWAKE
C
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WAKCRD / XW(133,24),     YW(133,24),     ZW(133,24)
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
C
C     ******************************************************************
C
C     BLADE SPACING AZIMUTH INCREMENT
C
      PSIBS = 2.0*PI/NBLDS
C
C     ROTATION TO POSITION OF FOLLOWING BLADE IS NEGATIVE
C
      DELPSI = -PSIBS
C
C     SWEEP BLADE AZIMUTH POSITION
C
      DO 300   IBLD = 2,NBLDS
C
C     SWEEP TRAILERS
C
      DO 200   J = 1,NP1
C
C     READ PREVIOUS BLADE'S TRAILER COORDINATES FROM DISK FILE
C
      IREC = (IBLD-2)*NP1+J
      READ(IOU13, REC = IREC) (XW(K,J),YW(K,J),ZW(K,J), K = 1,KMAX(J))
C
C     PLACE THIS TRAILER AT NEXT AZIMUTH
C
      DO 100   K = 1,KMAX(J)
      CALL ROTATE(XW(K,J),YW(K,J),DELPSI)
  100 CONTINUE
C
C     WRITE TO DISK FILE
```

```
C
      IREC = (IBLD-1)*NP1+J
      WRITE(IOU13, REC = IREC) (XW(K,J),YW(K,J),ZW(K,J), K = 1,KMAX(J))
C
  200 CONTINUE
C
  300 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT04(WAKPRM)
C
C     ADJUSTS REFERENCE WAKE PARAMETERS BASED ON THE WORKING STATE OF
C     THE ROTOR
C
C     ******************************************************************
C
      SUBROUTINE WAKPRM
C
      COMMON/ CTCON  / CTRIN,  CTROUT, CTRUPD
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ WRATES / AKZW,   AKZI,   AKZ,    AKR,    A,      AKB,
     1                 CTRW,   AKZO,   ALAMO,  AKRO,   VAXB,   VNUB
C
C     ******************************************************************
C
C     CALCULATE MOMENTUM INDUCED VELOCITY RATIO
C
      CTRW  =  CTRIN
      SGNCT =  SIGN(1.0,CTRW)
      VNUO  = OMEGAR*SGNCT*SQRT(ABS(CTRW)/2.0)
      VAXB  = VWO/VNUO
      AKAXB = VAXB/SQRT(2.0)
C
      IF(VAXB.GE.2.0) GO TO 100
      IF(VAXB.GE.1.5) GO TO 200
      IF(VAXB.GE.0.0) GO TO 300
      GO TO 400
C
  100 CONTINUE
C
C     WINDMILL BRAKE STATE
C
      VNUB = -VAXB/2.0+SQRT((VAXB/2.0)**2-1)
      AKB  =  AKAXB/2.0+SQRT((AKAXB/2.0)**2+1.0-SQRT(2.0))
      GO TO 500
C
  200 CONTINUE
C
C     VORTEX RING STATE, PAST REGION OF ROUGHNESS
C
      VNUB = 3.0*VAXB-7.0
      AKB  = VAXB+VNUB
      GO TO 500
C
  300 CONTINUE
```

```
C
C     VORTEX RING STATE, THROUGH REGION OF ROUGHNESS
C
      VNUB = -VAXB-1.0
      AKB  =  VAXB+VNUB
      GO TO 500
C
  400 CONTINUE
C
C     NORMAL WORKING STATE
C
      VNUB = -VAXB/2.0-SQRT((VAXB/2.0)**2+1.0)
      AKB  =  AKAXB/2.0-SQRT((AKAXB/2.0)**2+1.0)
C
  500 CONTINUE
C
C     SET REFERENCE WAKE PARAMETERS
C
      AKZO  = -SGNCT*SQRT(ABS(CTRW))
      ALAMO =  4.0*SQRT(ABS(CTRW))
      AKRO  =  ALAMO/AKZO
C
C     MODIFY REFERENCE VALUES BY VELOCITY RATIO
C
      AKZW =  VWO/OMEGAR
      AKZ  = -AKZO*AKB
      AKZI =  AKZ-AKZW
C
      AKR = ABS(AKRO/AKB)
      AKR = SIGN(AKR,AKZ)
      A   = 1.0+0.22*SIGN(1.0,AKZ)
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT04(WAKUPD)
C
C     CHECKS WAKE PARAMETER CONVERGENCE AND CALCULATES NEW VALUES BASED
C     ON MOST RECENT ITERATION
C
C     ******************************************************************
C
      SUBROUTINE WAKUPD
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
      COMMON/ CTCON  / CTRIN,  CTROUT, CTRUPD
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ WRATES / AKZW,   AKZI,   AKZ,    AKR,    A,      AKB,
     1                 CTRW,   AKZO,   ALAMO,  AKRO,   VAXB,   VNUB
C
C     ******************************************************************
C
C     SAVE CURRENT VALUES
C
      AKZ1   = AKZW+AKZI
      AKR1   = AKR
      CT1    = CTRIN
      CTRUPD = (CTRIN+CTROUT)/2.0
C
C     GET NEW VALUES
C
      CTRIN = CTROUT
C
      CALL WAKPRM
C
      IF(IPRT6.EQ.1) CALL OUT6
C
      AKZ2 = AKZW+AKZI
      AKR2 = AKR
C
C     CHECK CONVERGENCE
C
      IWFLAG = 0
      TOLMAX = 0.05
      IF(TOL3.GT.0.0) TOLMAX = TOL3
      TOLKZ  = ABS((AKZ2-AKZ1)/AKZ2)
      TOLKR  = ABS((AKR2-AKR1)/AKR2)
      IF(TOLKZ.GT.TOLMAX) IWFLAG = 1
      CTRIN = CTRUPD
C
      CALL WAKPRM
C
```

```
      IF(IPRT6.EQ.1) CALL OUT6                                  00000530
C                                                               00000540
      RETURN                                                    00000550
      END                                                       00000560
```

```
C     ******************************************************************
C
C     LSWT04(ROLNDX)
C
C     UPDATES WAKE SHEET INDICES FOR COMPATABILITY WITH RADIAL
C     CIRCULATION DISTRIBUTION
C
C     ******************************************************************
C
      SUBROUTINE ROLNDX
C
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
C
C     ******************************************************************
C
C     SEARCH FROM TIP INBOARD FOR FIRST MAXIMUM IN BOUND CIRCULATION
C
      JMAX   = NP
C
      DO 100  JJ = 2,NP
      J = NP-JJ+1
      IF(BCIRC(J).LT.BCIRC(J+1)) GO TO 150
      JMAX = J
  100 CONTINUE
C
  150 CONTINUE
C
C     GET CORRESPONDING SEGMENT
C
      DO 200  JSEG = 1,NSGMNT
      IF((JCGLIM(JSEG,1).LE.JMAX) .AND. (JMAX.LE.JCGLIM(JSEG,2)))
     1    JSEGM = JSEG
  200 CONTINUE
C
C     INBOARD INDEX OF OUTBOARD SHEET CORRESPONDS TO OUTBOARD
C     GRIDLINE INDEX OF THE JMAX PANEL
C
      JROLL = (JMAX-JCGLIM(JSEGM,1)+1) + JGLIM(JSEGM,1)
C
C     SET CONVERGENCE STATUS FLAG
```

```
C
      IF(JROLL.NE.JS(3,1)) IRFLAG = 1
C
C     RETURN IF NO CHANGE IN ROLL-UP
C
      IF(IRFLAG.EQ.0) GO TO 400
C
C     ADJUST OUTBOARD SHEET INDEX
C
      JS(3,1) = JROLL
C
C     ADJUST MID-SHEET OUTER BOUNDARY IF OPTIONED
C
      IF(JS(2,2).EQ.0) GO TO 300
      JS(2,2) = JROLL-1
C
C     MAKE SURE AT LEAST ONE FILAMENT IS LEFT
C
      IF(JS(2,2).GE.JS(2,1)) GO TO 400
      IF(JS(1,2).EQ.0) GO TO 250
      JS(2,1) = 0
      JS(2,2) = 0
      GO TO 300
  250 CONTINUE
      JS(2,1) = 1
      JS(2,2) = 1
      JS(3,1) = JS(1,2) + 1
      GO TO 400
C
  300 CONTINUE
C
C     OTHERWISE, ADJUST INBOARD SHEET OUTER BOUNDARY
C
      JS(1,2) = JROLL-1
      IF(JS(1,2).GE.JS(1,1)) GO TO 400
      JS(1,1) = 1
      JS(1,2) = 1
      JS(3,1) = JS(1,2) + 1
C
  400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT05(READIN)
C
C     DIRECTS ALL PROGRAM INPUT
C
C     ******************************************************************
C
      SUBROUTINE READIN
C
      CHARACTER CASEID*72
C
      COMMON/ CASE   / IPOINT, ALPHAD, VWOD,   IVDIM,  VWFAC,  PSIRD,
     1                 CASEID
      COMMON/ CGRID  / NGRID(4),       XG(6),  YG(24), XCG(5)
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ CONFIG / XOCREF(4),      YSEG(4,2),      JCHORD(4,2),
     1                 YCHORD(24),     CHORD(24),      JTWIST(4,2),
     2                 YTWIST(24),     TWIST(24),      JXAOFF(4,2),
     3                 YXAOFF(24),     XAOFF(24),      JAFT(4,2),
     4                 YAFT(24,2),     IDAF(24),       XTRAN(4),
     5                 YTRAN(4),       ZTRAN(4),       XROT(4),
     6                 YROT(4),        ZROT(4)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
      COMMON/ CTCON  / CTRIN,  CTROUT, CTRUPD
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ SWEEP  / ALPSIN, DALP,   NALP,   VWOIN,  DVWO,   NVWO,
     1                 PSIRIN, DPSIR,  NPSIR
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
C
C     ******************************************************************
C
      REWIND IOU5
C
C     ------------------------
C     RECORD SET 1, CASE SETUP
```

```
C     ------------------------
C
      READ(IOU5,10) CASEID
   10 FORMAT(A72)
C
C     PRINT SWITCHES
C
      READ (IOU5,11) APRT1,APRT2,APRT3,APRT4,APRT5,APRT6,APRT7
      READ (IOU5,11) APRT8
   11 FORMAT(7F10.0)
      IPRT1 = IFIX(APRT1+0.1)
      IPRT2 = IFIX(APRT2+0.1)
      IPRT3 = IFIX(APRT3+0.1)
      IPRT4 = IFIX(APRT4+0.1)
      IPRT5 = IFIX(APRT5+0.1)
      IPRT6 = IFIX(APRT6+0.1)
      IPRT7 = IFIX(APRT7+0.1)
      IPRT8 = IFIX(APRT8+0.1)
C
C     OUTPUT POINTER FOR TRACE OPTION
C
      IOU7 = IPRT7
      IF(IPRT7.EQ.1) IOU7 = 0
C
C     TOLERANCES
C
      READ(IOU5,11) TOL1,TOL2,TOL3
C
C     ITERATION CYCLE LIMITS
C
      READ(IOU5,11) AICYL1,AICYL2,AICYL3,AICYCL4
      ICYL1 = IFIX(AICYL1+0.1)
      ICYL2 = IFIX(AICYL2+0.1)
      ICYL3 = IFIX(AICYL3+0.1)
      ICYL4 = IFIX(AICYL4+0.1)
C
C     STREAM CONDITIONS AND PITCH SWEEP
C
      READ(IOU5,11) AISTDY,RPM,AMURAT,SIGMAP
      ISTDY = IFIX(AISTDY + 0.1)
      AMU   = 3.719E-07*AMURAT
      RHO   = 0.002378*SIGMAP
C
      READ(IOU5,11) VWOIN,DVWO,ANVWO,VDIM
      NVWO  = IFIX(ANVWO+0.1)
      IVDIM = IFIX(VDIM+0.1)
C
      READ(IOU5,11) ALPSIN,DALP,ANALP,THRUST
      ALPSIN = ALPSIN*RPD
      DALP   = DALP*RPD
      NALP   = ANALP+0.1
C
```

```
C     CONFIGURATION PARAMETERS
C
      READ(IOU5,11) ABLDS,AMP,ASGMNT,RADIUS
      NBLDS  = IFIX(ABLDS+0.1)
      MP     = IFIX(AMP+0.1)
      NSGMNT = IFIX(ASGMNT+0.1)
      OMEGAR = RPM*2*PI*RADIUS/60.0
      OMEGA  = OMEGAR/RADIUS
      CTRIN  = THRUST/(RHO*PI*RADIUS**2*OMEGAR**2)
      CTROUT = CTRIN
C
C     PUT VWOIN, DVWO INTO WORKING DIMENSIONS (FT/SEC)
C
C            IVDIM   INPUT DIMENSION
C              0       FT/SEC
C              1       M/SEC
C              2       MPH
C              3       TIP SPEED RATIO
C
      GO TO (50,51,52,53), IVDIM + 1
C
   50 CONTINUE
      VWFAC = 1.0
      GO TO 60
C
   51 CONTINUE
      VWFAC = 0.304800
      GO TO 60
C
   52 CONTINUE
      VWFAC = 15.0 / 22.0
      GO TO 60
C
   53 CONTINUE
      VWFAC = 1.0 / OMEGAR
C
C     CHECK FOR DVWO = 0.0, AND HANDLE, THEN RESTRICT TO ONE POINT
C
      IF(DVWO.LE.1.0E-06) NVWO = 1
      GO TO 70
C
   60 CONTINUE
C
      VWOIN = VWOIN / VWFAC
      DVWO  = DVWO  / VWFAC
C
   70 CONTINUE
C
C     WAKE LENGTH PARAMETERS
C
      READ(IOU5,11) REVS,ANWIPR,AIFWAK,PSIMR,PSIMT, AROLL
      NWIPR = IFIX(ANWIPR+0.1)
```

```
      IFWAK = IFIX(AIFWAK+0.1)
      PSIMR = -PSIMR*RPD
      PSIMT = -PSIMT*RPD
      IROLL = IFIX(AROLL+0.1)
C
      READ(IOU5,11) AJS11,AJS12,AJS21,AJS22,AJS31,AJS32
C
      JS(1,1) = IFIX(AJS11+0.1)
      JS(1,2) = IFIX(AJS12+0.1)
      IF(JS(1,1).EQ.0) JS(1,2) = 0
      IF(JS(1,2).EQ.0) JS(1,1) = 0
C
      JS(2,1) = IFIX(AJS21+0.1)
      JS(2,2) = IFIX(AJS22+0.1)
      IF(JS(2,1).EQ.0) JS(2,2) = 0
      IF(JS(2,2).EQ.0) JS(2,1) = 0
C
      JS(3,1) = IFIX(AJS31+0.1)
      JS(3,2) = IFIX(AJS32+0.1)
      IF(JS(3,1).EQ.0) JS(3,2) = 0
      IF(JS(3,2).EQ.0) JS(3,1) = 0
C
C     SET IROLL = 0 IF SELECTION NOT CORRECT
C
      IF(JS(3,2).EQ.0) IROLL = 0
      IF((JS(2,2).EQ.0) .AND. (JS(1,2).EQ.0)) IROLL = 0
C
C     --------------------------------
C     RECORD SET 2, SEGMENT DESCRIPTION
C     --------------------------------
C
      DO 500   JSEG = 1,NSGMNT
C
      READ(IOU5,11) ANPS,ANGRID,XOCREF(JSEG),YSEG(JSEG,1),YSEG(JSEG,2)
      READ(IOU5,11) XTRAN(JSEG),YTRAN(JSEG),ZTRAN(JSEG),
     1              XROT(JSEG),YROT(JSEG),ZROT(JSEG)
      NPS(JSEG)    = IFIX(ANPS+0.1)
      NGRID(JSEG)  = IFIX(ANGRID+0.1)
      YSEG(JSEG,1) = YSEG(JSEG,1)/RADIUS
      YSEG(JSEG,2) = YSEG(JSEG,2)/RADIUS
      XTRAN(JSEG)  = XTRAN(JSEG)/RADIUS
      YTRAN(JSEG)  = YTRAN(JSEG)/RADIUS
      ZTRAN(JSEG)  = ZTRAN(JSEG)/RADIUS
      XROT(JSEG)   = XROT(JSEG)*RPD
      YROT(JSEG)   = YROT(JSEG)*RPD
      ZROT(JSEG)   = ZROT(JSEG)*RPD
C
C     CHORD DISTRIBUTION
C
      READ(IOU5,11) ACHORD
      NCHORD = IFIX(ACHORD+0.1)
      J1 = 1
```

```
      IF(JSEG.GT.1) J1 = JCHORD(JSEG-1,2)+1
      J2 = J1+NCHORD-1
      READ(IOU5,11) (YCHORD(J), J = J1,J2)
      READ(IOU5,11) (CHORD(J),  J = J1,J2)
C
      DO 100   J = J1,J2
      YCHORD(J) = YCHORD(J)/RADIUS
      CHORD(J)  = CHORD(J)/RADIUS
  100 CONTINUE
C
      JCHORD(JSEG,1) = J1
      JCHORD(JSEG,2) = J2
C
C     TWIST DISTRIBUTION
C
      READ(IOU5,11) ATWIST
      NTWIST = IFIX(ATWIST+0.1)
      J1 = 1
      IF(JSEG.GT.1) J1 = JTWIST(JSEG-1,2)+1
      J2 = J1+NTWIST-1
      READ(IOU5,11) (YTWIST(J), J = J1,J2)
      READ(IOU5,11) (TWIST(J),  J = J1,J2)
C
      DO 200   J = J1,J2
      YTWIST(J) = YTWIST(J)/RADIUS
      TWIST(J)  = TWIST(J)*RPD
  200 CONTINUE
C
      JTWIST(JSEG,1) = J1
      JTWIST(JSEG,2) = J2
C
C     INPUT REFERENCE AXIS OFFSETS
C
      READ(IOU5,11) AXAOFF
      NXAOFF = IFIX(AXAOFF+0.1)
      J1 = 1
      IF(JSEG.GT.1) J1 = JXAOFF(JSEG-1,2)+1
      J2 = J1+NXAOFF-1
      READ(IOU5,11) (YXAOFF(J), J = J1,J2)
      READ(IOU5,11) (XAOFF(J),  J = J1,J2)
C
      DO 300   J = J1,J2
      YXAOFF(J) = YXAOFF(J)/RADIUS
      XAOFF(J)  = XAOFF(J)/RADIUS
  300 CONTINUE
C
      JXAOFF(JSEG,1) = J1
      JXAOFF(JSEG,2) = J2
C
C     AIRFOIL TABLE IDENTIFIERS
C
      READ(IOU5,11) ANAFT
```

```
      NAFT = IFIX(ANAFT+0.1)
      J1 = 1
      IF(JSEG.GT.1) J1 = JAFT(JSEG-1,2)+1
      J2 = J1+NAFT-1
c
      DO 400   J = J1,J2
      READ(IOU5,11) AIDAF,YAFT1,YAFT2
      IDAF(J)   = IFIX(AIDAF+0.1)
      YAFT(J,1) = YAFT1/RADIUS
      YAFT(J,2) = YAFT2/RADIUS
  400 CONTINUE
c
      JAFT(JSEG,1) = J1
      JAFT(JSEG,2) = J2
C
  500 CONTINUE
C
C     BYPASS RECORD SET 3 IF IN STEADY MODE
C
      IF(ISTDY.EQ.0) GO TO 800
C
C     ----------------------------
C     RECORD SET 3, NON-STEADY MODE
C     ----------------------------
C
      READ(IOU5,11) PSIRIN,DPSIR,APSIR
      PSIRIN = PSIRIN*RPD
      DPSIR  = DPSIR*RPD
      NPSIR  = IFIX(APSIR+0.1)
C
C     SHAFT AXIS ORIENTATION AND WIND PROFILE
C
      READ(IOU5,11) ALPS1,ALPS2,HO,E,ATSTYP
      ALPS1 = ALPS1*RPD
      ALPS2 = ALPS2*RPD
      ITSTYP = IFIX(ATSTYP+0.1)
      GO TO (552,550,551), ITSTYP+1
C
  550 CONTINUE
C
C     TOWER SHADOW EQUATION, ITSTYP = 1
C
      READ(IOU5,11) TO,TP,TPO,TPS,TK
      TPO = TPO*RPD
      TPS = TPS*RPD
      GO TO 552
C
  551 CONTINUE
C
C     TOWER SHADOW TABLE, ITSTYP = 2
C
      READ(IOU5,11) AXTS,AYTS
```

```
      NXTS = IFIX(AXTS+0.1)
      NYTS = IFIX(AYTS+0.1)
      CALL REDTAB(TSTAB(1),NXTS,NYTS)
C
  552 CONTINUE
C
C     FLAPPING AND PITCH BLADE MOTIONS
C
      READ(IOU5,11) AB(1,2),AB(1,1)
      NBH = IFIX(AB(1,2)+0.1)
      READ(IOU5,11) (AB(J,1), J = 2,NBH+1)
      READ(IOU5,11) (AB(J,2), J = 2,NBH+1)
      AB(1,1) = AB(1,1)*RPD
C
      DO 600   J = 2,NBH+1
      AB(J,1) = AB(J,1)*RPD
      AB(J,2) = AB(J,2)*RPD
  600 CONTINUE
C
      READ(IOU5,11) AT(1,2),AT(1,1)
      NTH = IFIX(AT(1,2)+0.1)
      READ(IOU5,11) (AT(J,1), J = 2,NTH+1)
      READ(IOU5,11) (AT(J,2), J = 2,NTH+1)
      AT(1,1) = AT(1,1)*RPD
C
      DO 700   J = 2,NTH+1
      AT(J,1) = AT(J,1)*RPD
      AT(J,2) = AT(J,2)*RPD
  700 CONTINUE
C
  800 CONTINUE
C     ---------------------------
C     RECORD SET 4, AIRFOIL TABLES
C     ---------------------------
C
      CALL READAF
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT05(READAF)
C
C     DOES PREPARATION FOR READING AIRFOIL TABLES AND CALLS REDTAB TO
C     READ TABLE ENTRIES
C
C     ******************************************************************
C
      SUBROUTINE READAF
C
      CHARACTER*62 TITLE(5)
C
      COMMON/ AFTABL / NAFT,   IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     READ NUMBER OF TABLES TO BE INPUT, TYPE OF EXTENSION TO USE
C
      READ(IOU5,11) ANAFT,AFTEX
   11 FORMAT(7F10.0)
      NAFT   = IFIX(ANAFT+0.1)
      IAFTEX = IFIX(AFTEX+0.1)
C
C     LOOP TO READ TABLES
C
      DO 400   J = 1,NAFT
C
      READ(IOU5,13) AIDAFT, TITLE(J)
   13 FORMAT(F10.0,A62)
      READ(IOU5,11) ANRCL,ANACL,ANRCD,ANACD,ANRCM,ANACM
C
      IDAFT(J) = IFIX(AIDAFT+0.1)
      NRCL(J)  = IFIX(ANRCL+0.1)
      NACL(J)  = IFIX(ANACL+0.1)
      NRCD(J)  = IFIX(ANRCD+0.1)
      NACD(J)  = IFIX(ANACD+0.1)
      NRCM(J)  = IFIX(ANRCM+0.1)
      NACM(J)  = IFIX(ANACM+0.1)
C
      READ(IOU5,11) AFTOC(J),STP1(J),STP2(J),STP3(J),STP4(J),STP5(J),
     1              STP6(J)
C
C     READ LIFT TABLE
C
```

```
      CALL REDTAB(CLTAB(1,J),NRCL(J),NACL(J))
C
C     SCALE INPUT REYNOLD'S NUMBERS
C
      DO 100   JJ = 1,NRCL(J)
      CLTAB(JJ,J)=CLTAB(JJ,J)*1.0E+06
  100 CONTINUE
C
C     READ DRAG TABLE
C
      CALL REDTAB(CDTAB(1,J),NRCD(J),NACD(J))
C
      DO 200   JJ = 1,NRCD(J)
      CDTAB(JJ,J)=CDTAB(JJ,J)*1.0E+06
  200 CONTINUE
C
C     READ PITCHING MOMENT TABLE
C
      CALL REDTAB(CMTAB(1,J),NRCM(J),NACM(J))
C
      DO 300   JJ = 1,NRCM(J)
      CMTAB(JJ,J)=CMTAB(JJ,J)*1.0E+06
  300 CONTINUE
C
  400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT05(REDTAB)
C
C     READS BIVARIENT TABLE ENTRIES AND ARRANGES THEM IN INTERNAL
C     STORAGE FORMAT
C
C     ******************************************************************
C
      SUBROUTINE REDTAB(TABLE,NCOL,NROW)
C
      DIMENSION TABLE(1)
C
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
      DATA ISIX/6/, IFIVE/5/
C
C     ******************************************************************
C
C     READ COLUMN VALUES INTO TABLE(1->NCOL)
C
      READ(IOU5,10) (TABLE(IS), IS = 1,NCOL)
   10 FORMAT(10X , 6F10.0)
C
      DO 200   I = 1,NROW
C
C     ICOUNT COUNTS THE NUMBER OF CARDS CONTAINING A ROW OF TABLE VALUES
C
      ICOUNT = 0
C
C     ISROW IS STORAGE LOCATION FOR ROW VALUE IN 'TABLE'
C
      ISROW = NCOL+I
C
C     IMAX IS MAXIMUM SUBSCRIPT IN 'TABLE' FOR THIS ROW'S VALUES
C     ISCOL1 IS FIRST SUBSCRIPT AND ISCOL2 IS LAST SUBSCRIPT FOR
C     TABLE VALUES READ FROM EACH CARD
C
      ISCOL1 = NCOL+NROW+(I-1)*NCOL+1
      IMAX   = ISCOL1+NCOL-1
      ISCOL2 = MIN0(IMAX,ISCOL1+IFIVE)
C
C     READ ROW VALUE INTO TABLE(NCOL+1->NCOL+NROW) AND TABLE VALUE INTO
C     TABLE(NCOL+NROW+1->NCOL+NROW+NCOL*NROW)
C
      READ(IOU5,12) TABLE(ISROW),(TABLE(IS), IS = ISCOL1,ISCOL2)
   12 FORMAT(7F10.0)
C
  100 CONTINUE
C
      ICOUNT = ICOUNT+1
```

```
C
C     IF MORE TABLE VALUES FOR THIS ROW ARE TO BE READ, THEN CONTINUE
C
      IF(NCOL.LE.ISIX*ICOUNT) GO TO 200
      ISCOL1 = ISCOL2+1
      ISCOL2 = MIN0(IMAX,ISCOL1+IFIVE)
      READ(IOU5,10) (TABLE(IS), IS = ISCOL1,ISCOL2)
      GO TO 100
C
  200 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT06(OUT1)
C
C     REPORTS BLADE REFERENCE GEOMETRY
C
C     ******************************************************************
C
      SUBROUTINE OUT1(ITBLD)
C
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLCRD / XS(6,24),       YS(6,24),       ZS(6,24),
     1                 XCOL(5,20),     YCOL(5,20),     ZCOL(5,20)
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
      ICASE = 1
      IF(ITBLD.GE.1) ICASE = 2
      CALL HEADER(ICASE,IOU6)
C
      IF(ITBLD.EQ.0) WRITE(IOU6,10)
   10 FORMAT('0',4X,'REFERENCE BLADE SURFACE PANEL COORDINATES'/
     1       ' ',4X,41('-'))
C
      IF(ITBLD.GT.0) WRITE(IOU6,12) ITBLD
   12 FORMAT('0',4X,'BLADE ',I2,' SURFACE PANEL COORDINATES'/
     1       ' ',4X,37('-'))
C
      J1 = 1
      J2 = NP1
      IF(NP1.GT.11) J2 = 11
C
 1000 CONTINUE
C
      IF(J2-J1.GT.10) J2 = J1+10
C
      WRITE(IOU6,14) (J, J = J1,J2)
   14 FORMAT('0',4X,'I,J',I8,10I10)
C
      DO 1100   I = 1,MP1
C
      WRITE(IOU6,16) I,(XS(I,J), J = J1,J2)
   16 FORMAT('0',4X,I2,1X,'X',11F10.6)
C
      WRITE(IOU6,18)   (YS(I,J), J = J1,J2)
   18 FORMAT(' ',7X,'Y',11F10.6)
```

```
C
      WRITE(IOU6,20)   (ZS(I,J), J = J1,J2)
   20 FORMAT(' ',7X,'Z',11F10.6)
C
 1100 CONTINUE
C
      JTEMP = J2
      J1    = J2+1
      J2    = J1+10
      IF(J2.GT.NP1) J2 = NP1
      IF(JTEMP.LT.NP1) GO TO 1000
C
C     COLLOCATION POINT COORDINATES
C
      CALL HEADER(ICASE,IOU6)
C
      IF(ITBLD.EQ.0) WRITE(IOU6,22)
   22 FORMAT('0',4X,'REFERENCE BLADE COLLOCATION POINT COORDINATES'/
     1       ' ',4X,45('-'))
C
      IF(ITBLD.GT.0) WRITE(IOU6,24) ITBLD
   24 FORMAT('0',4X,'BLADE ',I2,' COLLOCATION POINT COORDINATES'/
     1       ' ',4X,41('-'))
C
      J1 = 1
      J2 = NP
C
 1200 CONTINUE
C
      IF(J2-J1.GT.9) J2 = J1+9
C
      WRITE(IOU6,26) (J, J = J1,J2)
   26 FORMAT('0',4X,'I,J',I8,9I10)
C
      DO 1300   I = 1,MP
C
      WRITE(IOU6,28) I,(XCOL(I,J), J=J1,J2)
   28 FORMAT('0',4X,I2,1X,'X',10F10.6)
C
      WRITE(IOU6,30)   (YCOL(I,J), J=J1,J2)
   30 FORMAT(' ',7X,'Y',10F10.6)
C
      WRITE(IOU6,32)   (ZCOL(I,J), J=J1,J2)
   32 FORMAT(' ',7X,'Z',10F10.6)
C
 1300 CONTINUE
C
      JTEMP = J2
      J1 = J2+1
      J2 = J1+9
      IF(J2.GT.NP) J2 = NP
      IF(JTEMP.LT.NP) GO TO 1200
```

```
C                                                                       00001050
      RETURN                                                            00001060
      END                                                               00001070
```

```
C     ******************************************************************
C
C     LSWT06(OUT2)
C
C     REPORTS WAKE COORDINATES
C
C     ******************************************************************
C
      SUBROUTINE OUT2(ITBLD,PSIR)
C
      DIMENSION PSI(133),RW(133)
C
      COMMON/ CONSTS / PI,     RPD
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WAKCRD / XW(133,24),     YW(133,24),     ZW(133,24)
      COMMON/ WAKNDX / REVS,           NWIPR,          NWI,
     1                 KMAX(24),       IFWAK,          JS(3,2),
     2                 PSIMR,          PSIMT,          IROLL
C     ******************************************************************
C
C     LOAD WAKE COORDINATES FOR THIS BLADE INTO BUFFER
C
      DO 100  J = 1,NP1
C
C     LOAD THE BUFFER
C
      IREC = (ITBLD-1)*NP1+J
      READ(IOU13, REC=IREC) (XW(K,J),YW(K,J),ZW(K,J), K = 1,KMAX(J))
C
  100 CONTINUE
C
      CALL HEADER(2,IOU6)
C
      WRITE(IOU6,10) ITBLD
   10 FORMAT('0',4X,'BLADE ',I2,' WAKE FILAMENT COORDINATES'/
     1       ' ',4X,34('-'))
C
       SITB  = PSIR - FLOAT(ITBLD-1)*(2.0*PI/FLOAT(NBLDS))
      PSIRD  = PSIR/RPD
      PSITBD = PSITB/RPD
C
C     SWEEP WAKE FILAMENTS
C
      DO 400   J = 1,NP1
C
C     CALCULATE CYLINDRICAL COORDINATES
C
```

```
      PSIM1 = 0.0
C
      DO 200  K=1,KMAX(J)
C
C     FIRST REFERENCE TO TARGET BLADE
C
      XTW   = XW(K,J)
      YTW   = YW(K,J)
      PSITW = -PSITB
      CALL ROTATE(XTW,YTW,PSITW)
      RW(K)  = SQRT(XW(K,J)**2+YW(K,J)**2)
      PSITWD = ATAN2(-XTW,YTW)/RPD
C
C     PUT IN 0 TO -360 DEG RANGE
C
      IF(PSITWD.GT.0.0) PSITWD = PSITWD-360.0
C
C     ADD IN EARLIER REVS
C
      IREVS  = IFIX(ABS(PSIM1/360.0))
      PSITWD = PSITWD+IREVS*(-360.0)
      IF(PSITWD.GT.PSIM1) PSITWD = PSITWD - 360.0
      PSIM1  = PSITWD
C
C     REFERENCE BACK TO ROTOR AZIMUTH
C
      PSI(K) = PSITWD + PSITBD
C
  200 CONTINUE
C
C     REPORT OUT THIS FILAMENT
C
      WRITE(IOU6,11) J
   11 FORMAT('0',7X,'FILAMENT NUMBER ',I2,)
C
      K1 = 1
      K2 = KMAX(J)
      IF(K2-K1.GT.10) K2 = K1+10
C
  300 CONTINUE
C
      WRITE(IOU6,12) (K, K = K1,K2)
   12 FORMAT('0',7X,'K',11(5X,I2,3X))
C
      WRITE(IOU6,13) (PSI(K), K = K1,K2)
   13 FORMAT(' ',5X,'PSI',11F10.3)
C
      WRITE(IOU6,14) (RW(K), K = K1,K2)
   14 FORMAT(' ',7X,'R',11F10.6)
C
      WRITE(IOU6,15) (XW(K,J), K = K1,K2)
   15 FORMAT(' ',7X,'X',11F10.6)
```

```
C
      WRITE(IOU6,16) (YW(K,J), K = K1,K2)
   16 FORMAT(' ',7X,'Y',11F10.6)
C
      WRITE(IOU6,17) (ZW(K,J), K = K1,K2)
   17 FORMAT(' ',7X,'Z',11F10.6)
C
      WRITE(IOU6,18)
   18 FORMAT('0')
C
C     RESET AND MAKE ANOTHER PASS IF REQUIRED
C
      KTEMP = K2
      K1    = K2+1
      K2    = K1+10
      IF(K2.GT.KMAX(J)) K2 = KMAX(J)
      IF(KTEMP.LT.KMAX(J)) GO TO 300
C
  400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT06(OUT3)
C
C     REPORTS BLADE ELEMENT AERODYNAMIC CHARACTERISTICS
C
C     ******************************************************************
C
      SUBROUTINE OUT3(ITBLD,IOUTO)
C
      COMMON/ ALSLP  / ALS(20),        ALSNEW(20),     ALSOLD(20)
      COMMON/ BELOAD / DFX(20),        DFXI(20),       DFXP(20),
     1                 DFZ(20),        DFZI(20),       DFZP(20),
     2                 DQ(20),         DQI(20),        DQP(20),
     3                 DPM(20)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),
     2                 TWSTCG(20)
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CVLCTY / VFS1(5,20),     VFS2(5,20),     VFS3(5,20),
     1                 VFS1M(20),      VFS2M(20),      VFS3M(20),
     2                 VI1(5,20),      VI2(5,20),      VI3(5,20),
     3                 VI1M(20),       VI2M(20),       VI3M(20),
     4                 VREL(20),       VIND(20),       QREL(20)
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
C
C     ******************************************************************
C
C     CHANGE DIMENSIONS FOR OUTPUT REPORT
C
      DO 100   J = 1,NP
      ALPHAG(J) = ALPHAG(J)/RPD
      ALPHAE(J) = ALPHAE(J)/RPD
      ALPHAW(J) = ALPHAW(J)/RPD
      ALPHAN(J) = ALPHAN(J)/RPD
      ALPHAI(J) = ALPHAI(J)/RPD
      AZLLC(J)  = AZLLC(J)/RPD
      CHDCGL(J) = CHDCGL(J)*RADIUS
      DBES(J)   = DBES(J)*RADIUS
```

```
      CLAMQC(J) = ACOS(CLAMQC(J))/RPD
      CLAMLE(J) = ACOS(CLAMLE(J))/RPD
  100 CONTINUE
C
C     BLADE ELEMENT AERODYNAMIC CHARACTERISTIC REPORT
C
      J1 = 1
      J2 = NP
C
  200 CONTINUE
C
      IF(J2-J1.GT.7) J2 = J1+7
C
      CALL HEADER(2,IOUTO)
C
      WRITE(IOUTO,1000) ITBLD
 1000 FORMAT('0',4X,'BLADE ',I2,' ELEMENT CHARACTERISTICS'/
     1       ' ',4X,32('-'))
C
C     ELEMENT GEOMETRY
C
      WRITE(IOUTO,2000)
 2000 FORMAT('0',4X,'GEOMETRY')
C
      WRITE(IOUTO,2001) (J, J = J1,J2)
 2001 FORMAT('0',4X,'ELEMENT NUMBER',16X,8I11)
C
      WRITE(IOUTO,2002) (RBE(J), J = J1,J2)
 2002 FORMAT(' ',4X,'ELE CENTER STATION',16X,1P8E11.3)
C
      WRITE(IOUTO,2003) (CHDCGL(J), J = J1,J2)
 2003 FORMAT(' ',4X,'CHORD',20X,'FT',7X,1P8E11.3)
C
      WRITE(IOUTO,2004) (DBES(J), J = J1,J2)
 2004 FORMAT(' ',4X,'ELEMENT SPAN',13X,'FT',7X,1P8E11.3)
C
      WRITE(IOUTO,2005) (CLAMLE(J), J = J1,J2)
 2005 FORMAT(' ',4X,'LEADING EDGE SWEEP',7X,'DEG',6X,1P8E11.3)
C
      WRITE(IOUTO,2006) (CLAMQC(J), J = J1,J2)
 2006 FORMAT(' ',4X,'AERODYNAMIC SWEEP',8X,'DEG',6X,1P8E11.3)
C
C     ANGLES
C
      WRITE(IOUTO,3000)
 3000 FORMAT('0',4X,'ANGLES OF ATTACK')
C
      WRITE(IOUTO,3001) (ALPHAG(J), J = J1,J2)
 3001 FORMAT('0',4X,'GEOMETRIC PITCH',10X,'DEG',6X,1P8E11.3)
C
      WRITE(IOUTO,3002) (AZLLC(J), J = J1,J2)
 3002 FORMAT(' ',4X,'ZERO LIFT LINE',11X,'DEG',6X,1P8E11.3)
```

```
C
      WRITE(IOUTO,3004) (ALPHAE(J), J = J1,J2)
 3004 FORMAT(' ',4X,'EFFECTIVE ANGLE',10X,'DEG',6X,1P8E11.3)
C
      WRITE(IOUTO,3005) (ALPHAW(J), J = J1,J2)
 3005 FORMAT(' ',4X,'WIND ANGLE',15X,'DEG',6X,1P8E11.3)
C
      WRITE(IOUTO,3006) (ALPHAN(J), J = J1,J2)
 3006 FORMAT(' ',4X,'INFLOW ANGLE',13X,'DEG',6X,1P8E11.3)
C
      WRITE(IOUTO,3007) (ALPHAI(J), J = J1,J2)
 3007 FORMAT(' ',4X,'INDUCED ANGLE',12X,'DEG',6X,1P8E11.3)
C
C     VELOCITIES
C
      WRITE(IOUTO,4000)
 4000 FORMAT('0',4X,'VELOCITIES')
C
      WRITE(IOUTO,4001) (VIND(J), J = J1,J2)
 4001 FORMAT('0',4X,'INDUCED VELOCITY',9X,'FT/SEC',3X,1P8E11.3)
C
      WRITE(IOUTO,4002) (VREL(J), J = J1,J2)
 4002 FORMAT(' ',4X,'RELATIVE VELOCITY',8X,'FT/SEC',3X,1P8E11.3)
C
      WRITE(IOUTO,4003) (RE(J), J = J1,J2)
 4003 FORMAT(' ',4X,'REYNOLDS NUMBER',19X,1P8E11.3)
C
      WRITE(IOUTO,4004) (QREL(J), J = J1,J2)
 4004 FORMAT(' ',4X,'RELATIVE Q',15X,'PSF',6X,1P8E11.3)
C
C     LIFT, DRAG AND MOMENT COEFFICIENTS
C
      WRITE(IOUTO,5000)
 5000 FORMAT('0',4X,'AERODYNAMIC COEFFICIENTS')
C
      WRITE(IOUTO,5001) (CLT(J), J = J1,J2)
 5001 FORMAT('0',4X,'ELE LIFT COEF',21X,1P8E11.3)
C
      WRITE(IOUTO,5002) (CDT(J), J = J1,J2)
 5002 FORMAT(' ',4X,'ELE DRAG COEF',21X,1P8E11.3)
C
      WRITE(IOUTO,5003) (CMT(J), J = J1,J2)
 5003 FORMAT(' ',4X,'ELE MOMENT COEF',19X,1P8E11.3)
C
C     ELEMENT FORCE LOADINGS
C
      WRITE(IOUTO,6000)
 6000 FORMAT('0',4X,'FORCE LOADING')
C
      WRITE(IOUTO,6001) (DFX(J), J = J1,J2)
 6001 FORMAT(' ',4X,'ELE X FORCE',14X,'LBS/FT',3X,1P8E11.3)
C
```

```
      WRITE(IOUTO,6002) (DFXI(J), J = J1,J2)
 6002 FORMAT(' ',4X,'ELE IND X FORCE',10X,'LBS/FT',3X,1P8E11.3)
C
      WRITE(IOUTO,6003) (DFXP(J), J = J1,J2)
 6003 FORMAT(' ',4X,'ELE PRO X FORCE',10X,'LBS/FT',3X,1P8E11.3)
C
      WRITE(IOUTO,6004) (DFZ(J), J = J1,J2)
 6004 FORMAT(' ',4X,'ELE Z FORCE',14X,'LBS/FT',3X,1P8E11.3)
C
      WRITE(IOUTO,6005) (DFZI(J), J = J1,J2)
 6005 FORMAT(' ',4X,'ELE IND Z FORCE',10X,'LBS/FT',3X,1P8E11.3)
C
      WRITE(IOUTO,6006) (DFZP(J), J = J1,J2)
 6006 FORMAT(' ',4X,'ELE PRO Z FORCE',10X,'LBS/FT',3X,1P8E11.3)
C
C     ELEMENT TORQUE LOADINGS
C
      WRITE(IOUTO,7000)
 7000 FORMAT('0',4X,'TORQUE LOADING')
C
      WRITE(IOUTO,7001) (DQP(J), J = J1,J2)
 7001 FORMAT('0',4X,'ELE PRO TORQUE',11X,'FT-LBS/FT',1P8E11.3)
C
      WRITE(IOUTO,7002) (DQI(J), J = J1,J2)
 7002 FORMAT(' ',4X,'ELE IND TORQUE',11X,'FT-LBS/FT',1P8E11.3)
C
      WRITE(IOUTO,7003) (DQ(J), J = J1,J2)
 7003 FORMAT(' ',4X,'ELE TORQUE',15X,'FT-LBS/FT',1P8E11.3)
C
C     PITCHING MOMENT LOADING (2D)
C
      WRITE(IOUTO,8000)
 8000 FORMAT('0',4X,'PITCHING MOMENT LOADING (2D)')
C
      WRITE(IOUTO,8001) (DPM(J), J = J1,J2)
 8001 FORMAT('0',4X,'ELE PITCH MOMENT',9X,'FT-LBS/FT',1P8E11.3)
C
C     LIFT CURVE SLOPES
C
      WRITE(IOUTO,9000)
 9000 FORMAT('0',4X,'LIFT SLOPE CONVERGENCE')
C
      WRITE(IOUTO,9001) (ALSOLD(J), J = J1,J2)
 9001 FORMAT('0',4X,'INPUT LIFT SLOPE',9X,'/RAD',5X,1P8E11.3)
C
      WRITE(IOUTO,9002) (ALSNEW(J), J = J1,J2)
 9002 FORMAT(' ',4X,'CL/ALPHAE',16X,'/RAD',5X,1P8E11.3)
C
C     CIRCULATION
C
      WRITE(IOUTO,10000)
10000 FORMAT('0',4X,'CIRCULATION PARAMETERS')
```

```
C
      WRITE(IOUTO,10001) (GO(J), J = J1,J2)
10001 FORMAT('0',4X,'CIRC SHAPE PARM',19X,1P8E11.3)
C
      WRITE(IOUTO,10002) (BCIRC(J), J = J1,J2)
10002 FORMAT(' ',4X,'BOUND CIRCULATION',8X,'FT**2/SEC',1P8E11.3)
C
      IF(J2.GE.NP) GO TO 300
C
      J1 = J2+1
      J2 = J1+7
      IF(J2.GT.NP) J2 = NP
      GO TO 200
C
  300 CONTINUE
C
C     RETURN TO WORKING DIMENSIONS
C
      DO 400   J = 1,NP
      ALPHAG(J) = ALPHAG(J)*RPD
      ALPHAE(J) = ALPHAE(J)*RPD
      ALPHAW(J) = ALPHAW(J)*RPD
      ALPHAN(J) = ALPHAN(J)*RPD
      ALPHAI(J) = ALPHAI(J)*RPD
      AZLLC(J)  = AZLLC(J)*RPD
      CHDCGL(J) = CHDCGL(J)/RADIUS
      DBES(J)   = DBES(J)/RADIUS
      CLAMQC(J) = COS(CLAMQC(J)*RPD)
      CLAMLE(J) = COS(CLAMLE(J)*RPD)
  400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT06(OUT4)
C
C     REPORTS BLADE PERFORMANCE CHARACTERISTICS
C
C     ******************************************************************
C
      SUBROUTINE OUT4(ITBLD,IOUTO)
C
      COMMON/ BLLOAD / CTRB,   CQRB,   CTWB,   CPWB,   BTHRST, BTORQ,
     1                 BTORQI, BTORQP, BPWR,   BPWRP,  BPWRI,  BPMOMT
      COMMON/ CONSTS / PI,     RPD
C
C     ******************************************************************
C
C     BLADE PERFORMANCE SUMMARY
C
      CALL HEADER(2,IOUTO)
C
      WRITE(IOUTO,1100) ITBLD
 1100 FORMAT('0',4X,'BLADE ',I2,' PERFORMANCE'/' ',4X,20('-'))
C
      WRITE(IOUTO,1101) BTHRST,CTRB,CTWB
 1101 FORMAT('0',4X,'THRUST',20X,'LBS',8X,1PE12.4/
     1       ' ',4X,'CT ROTOR',29X,1PE12.4/
     2       ' ',4X,'CT WT',32X,1PE12.4)
C
      WRITE(IOUTO,1102) BTORQ,CQRB,CPWB
 1102 FORMAT('0',4X,'TORQUE',20X,'FT-LBS',5X,1PE12.4/
     1       ' ',4X,'CQ ROTOR',29X,1PE12.4/
     2       ' ',4X,'CP WT',32X,1PE12.4)
C
      WRITE(IOUTO,1103) BPWR,BPWRP,BPWRI
 1103 FORMAT('0',4X,'POWER (BLADE)  ',11X,'KW',9X,1PE12.4/
     1       ' ',4X,'POWER (PROFILE)',11X,'KW',9X,1PE12.4/
     2       ' ',4X,'POWER (INFLOW) ',11X,'KW',9X,1PE12.4)
C
      WRITE(IOUTO,1104) BPMOMT
 1104 FORMAT('0',4X,'PITCH MOM (2D)',12X,'FT-LB',6X,1PE12.4)
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT06(OUT5)
C
C     REPORTS ROTOR PERFORMANCE CHARACTERISTICS
C
C     ******************************************************************
C
      SUBROUTINE OUT5
C
      COMMON/ CONSTS / PI,     RPD
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ RTLOAD / CTRR,   CQRR,   CTWR,   CPWR,   RTHRST, RTORQ,
     1                 RTORQP, RTORQI, RPWR,   RPWRP,  RPWRI,  CTR,
     2                 CQR,    CTW,    CPW,    THRST,  TORQ,   TORQP,
     3                 TORQI,  PWR,    PWRP,   PWRI,   RPMOMT, PMOMT
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     ROTOR PERFORMANCE SUMMARY
C
      CALL HEADER(2,IOU6)
C
      WRITE(IOU6,1200)
 1200 FORMAT('0',4X,'ROTOR PERFORMANCE'/
     1       ' ',4X,17('-'))
C
      WRITE(IOU6,1201) RTHRST,CTRR,CTWR
 1201 FORMAT('0',4X,'THRUST',20X,'LBS',8X,1PE12.4/
     1       ' ',4X,'CT ROTOR',29X,1PE12.4/
     2       ' ',4X,'CT WT',32X,1PE12.4)
C
      WRITE(IOU6,1202) RTORQ,CQRR,CPWR
 1202 FORMAT('0',4X,'TORQUE',20X,'FT-LBS',5X,1PE12.4/
     1       ' ',4X,'CQ ROTOR',29X,1PE12.4/
     2       ' ',4X,'CP WT',32X,1PE12.4)
C
      WRITE(IOU6,1203) RPWR,RPWRP,RPWRI
 1203 FORMAT('0',4X,'POWER (ROTOR)  ',11X,'KW',9X,1PE12.4/
     1       ' ',4X,'POWER (PROFILE)',11X,'KW',9X,1PE12.4/
     2       ' ',4X,'POWER (INFLOW) ',11X,'KW',9X,1PE12.4)
C
      IF(ISTDY.EQ.0) GO TO 1400
C
 1250 CONTINUE
C
      WRITE(IOU6,1300)
 1300 FORMAT('0',4X,'DISK AVERAGE PERFORMANCE'/
     1       ' ',4X,24('-'))
C
```

```
      WRITE(IOU6,1301) THRST,CTR,CTW
 1301 FORMAT('0',4X,'THRUST',20X,'LBS',8X,1PE12.4/
     1       ' ',4X,'CT ROTOR',29X,1PE12.4/
     2       ' ',4X,'CT WT',32X,1PE12.4)
C
      WRITE(IOU6,1302) TORQ,CQR,CPW
 1302 FORMAT('0',4X,'TORQUE',20X,'FT-LBS',5X,1PE12.4/
     1       ' ',4X,'CQ ROTOR',29X,1PE12.4/
     2       ' ',4X,'CP WT',32X,1PE12.4)
C
      WRITE(IOU6,1303) PWR,PWRP,PWRI
 1303 FORMAT('0',4X,'POWER (ROTOR)',13X,'KW',9X,1PE12.4/
     1       ' ',4X,'POWER (PRO)  ',13X,'KW',9X,1PE12.4/
     2       ' ',4X,'POWER (IND)  ',13X,'KW',9X,1PE12.4)
C
      WRITE(IOU6,1304) PMOMT
 1304 FORMAT('0',4X,'PITCH MOM (2D)',12X,'FT-LB',6X,1PE12.4)
C
 1400 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT06(OUT6)
C
C     REPORTS WAKE PARAMETERS AND WORK STATE
C
C     ******************************************************************
C
      SUBROUTINE OUT6
C
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
      COMMON/ WRATES / AKZW,   AKZI,   AKZ,    AKR,    A,      AKB,
     1                 CTRW,   AKZO,   ALAMO,  AKRO,   VAXB,   VNUB
C
C     ******************************************************************
C
      CALL HEADER(2,IOU6)
C
      WRITE(IOU6,10)
   10 FORMAT('0',4X,'WORKING STATE AND WAKE RATE PARAMETERS'/
     1       ' ',4X,38('-'))
C
C     CHECK FOR WORKING STATE
C
      IF(VAXB.GE.2.0) GO TO 100
      IF(VAXB.GE.1.5) GO TO 200
      IF(VAXB.GE.0.0) GO TO 300
      GO TO 400
C
  100 CONTINUE
      WRITE(IOU6,101)
  101 FORMAT('0',4X,'WINDMILL BRAKE STATE')
      GO TO 500
C
  200 CONTINUE
      WRITE(IOU6,201)
  201 FORMAT('0',4X,'VORTEX RING STATE, PAST REGION OF ROUGHNESS')
      GO TO 500
C
  300 CONTINUE
      WRITE(IOU6,301)
  301 FORMAT('0',4X,'VORTEX RING STATE, THROUGH REGION OF ROUGHNESS')
      GO TO 500
C
  400 CONTINUE
      WRITE(IOU6,401)
  401 FORMAT('0',4X,'NORMAL WORKING STATE')
C
  500 CONTINUE
C
      WRITE(IOU6,601) VAXB,VNUB
```

```
  601 FORMAT('0',6X,'REFERRED VELOCITIES,'/                          00000530
     1       ' ',8X,'AXIAL ONSET',1PE14.5/                           00000540
     2       ' ',8X,'INDUCED    ',1PE14.5)                           00000550
C                                                                    00000560
      WRITE(IOU6,602) CTRW,AKZO,ALAMO,AKRO                           00000570
  602 FORMAT('0',6X,'REFERENCE WAKE PARAMETERS,'/                    00000580
     1       ' ',8X,'CTRW       ',1PE14.5/                           00000590
     2       ' ',8X,'AKZO       ',1PE14.5/                           00000600
     3       ' ',8X,'ALAMO      ',1PE14.5/                           00000610
     4       ' ',8X,'AKRO       ',1PE14.5)                           00000620
C                                                                    00000630
      WRITE(IOU6,603) AKZW,AKZI,AKZ,AKR,A,AKB                        00000640
  603 FORMAT('0',6X,'MODIFIED WAKE PARAMETERS,'/                     00000650
     1       ' ',8X,'AKZW       ',1PE14.5/                           00000660
     2       ' ',8X,'AKZI       ',1PE14.5/                           00000670
     3       ' ',8X,'AKZ        ',1PE14.5/                           00000680
     4       ' ',8X,'AKR        ',1PE14.5/                           00000690
     5       ' ',8X,'A          ',1PE14.5/                           00000700
     6       ' ',8X,'AKB        ',1PE14.5)                           00000710
C                                                                    00000720
      RETURN                                                         00000730
      END                                                            00000740
```

```
C     ******************************************************************
C
C     LSWT06(OUT8)
C
C     REPORTS RECORD IMAGE OF INPUT FILE
C
C     ******************************************************************
C
      SUBROUTINE OUT8
C
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
      CALL HEADER(0,IOU6)
C
      WRITE(IOU6,10)
   10 FORMAT('0',4X,'RECORD IMAGE OF INPUT FILE'/
     1       ' ',4X,26('-')//)
C
      CALL APPEND(IOU5,IOU6,80,5)
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT06(HEADER)
C
C     PLACES BANNER AND CASE INFORMATION ON REPORTS
C
C       ICASE = 0  BANNER ONLY
C       ICASE = 1  BANNER AND CASE ID
C       ICASE = 2  BANNER, CASE ID, AND POINT INFORMATION
C
C     ******************************************************************
C
      SUBROUTINE HEADER(ICASE,IOUTO)
C
      CHARACTER TM     *36
      CHARACTER SPEC   *35
      CHARACTER STATUS *15
      CHARACTER CASEID *72
      CHARACTER DIM(4) *7
C
      COMMON/ CASE   / IPOINT, ALPHAD, VWOD,   IVDIM,  VWFAC,  PSIRD,
     1                 CASEID
C
      DATA TM     /'COMPUTATIONAL METHODOLOGY ASSOCIATES'/
      DATA SPEC   /'LSWT VERSION 1.00 RELEASED MAY 1986'/
      DATA STATUS /'SERI IL-5-05105'/
      DATA DIM    /'(FT/SEC',' (M/SEC','   (MPH','   (TSR'/
C
C     ******************************************************************
C
      WRITE(IOUTO,10) TM,SPEC,STATUS
   10 FORMAT('1',4X,A36,' *** ',A35,' *** ',A15,/' ',4X,122('-'))
C
      IF(ICASE.NE.1) GO TO 100
C
      WRITE(IOUTO,20) CASEID
   20 FORMAT('0',4X,A72)
      GO TO 200
C
  100 CONTINUE
C
      IF(ICASE.NE.2) GO TO 200
C
      WRITE(IOUTO,30) IPOINT,CASEID,ALPHAD,DIM(IVDIM+1),VWOD,PSIRD
   30 FORMAT('0',4X,'POINT ',I2,': ',A72/
     1       ' ',4X,'ROTOR PITCH      (DEG) ',F10.4,/
     2       ' ',4X,'WIND SPEED    ' ,A7,') ',F10.4,/
     3       ' ',4X,'ROTOR AZIMUTH    (DEG) ',F10.4)
C
  200 CONTINUE
C
      RETURN
```

```
      END                                                       00000530
```

```
C     ******************************************************************
C
C     LSWTO6(APPEND)
C     COPIES THE FILE ASSOCIATED WITH UNIT IOUFRM TO THE FILE ON UNIT
C     IOUTO AT ITS CURRENT POINTER POSITION.  BOTH FILES MUST BE
C     FORMATTED, SEQUENTIAL FILES.  'MRECL' IS THE MAXIMUM RECORD LENGTH
C     OF THE IOUFRM FILE.  TRAILING BLANKS ARE TRIMMED FROM THE RECORD
C     TO CONSERVE SPACE.  IF LPAD > 0 THEN THE RECORD IS PADDED WITH
C     LPAD LEADING BLANKS TO ACCOMODATE FORMAT CONTROL ON OUTPUT.
C     (MRECL+LPAD) MUST BE LE 132.
C
C     ******************************************************************
C
      SUBROUTINE APPEND(IOUFRM, IOUTO, MRECL,LPAD)
C
      CHARACTER IOBUFR(132) *1
C
C     ******************************************************************
C
C     RESTRICT VALUES FOR MAXIMUM RECORD LENGTH
C
      IF(MRECL+LPAD .GT. 132) GO TO 600
C
C     POSITION RECORD POINTER TO BEGINNING OF 'FROM' FILE
C
      REWIND IOUFRM
C
C     READ IOUFRM AND WRITE IOUTO UNTIL END OF FILE
C
  100 CONTINUE
C
      READ  (IOUFRM,10,END=600) (IOBUFR(I), I=1,MRECL)
   10 FORMAT(132A1)
C
C     GET NET LENGTH BY FINDING FIRST NON-BLANK CHARACTER FROM END
C
      IRECL = MRECL
C
      DO 200  I = 1, MRECL
      II = MRECL - I + 1
      IASCII = ICHAR(IOBUFR(II))
      IF(IASCII .EQ. 32) GO TO 200
      IRECL = II
      GO TO 300
  200 CONTINUE
C
  300 CONTINUE
C
C     PAD WITH LEADING BLANK IF REQUESTED, LPAD > 0
C
      IF(LPAD.EQ.0) GO TO 500
      DO 400 I = 1, IRECL
```

```
      II = IRECL - I + 1                                                00000530
      IOBUFR(II+LPAD) = IOBUFR(II)                                      00000540
  400 CONTINUE                                                          00000550
C                                                                       00000560
      DO 410 I=1, LPAD                                                  00000570
      IOBUFR(I) = ' '                                                   00000580
  410 CONTINUE                                                          00000590
      IRECL = IRECL + LPAD                                              00000600
C                                                                       00000610
  500 CONTINUE                                                          00000620
C                                                                       00000630
C     APPEND RECORD TO 'TO' FILE                                        00000640
C                                                                       00000650
      WRITE (IOUTO,10) (IOBUFR(I), I=1,IRECL)                           00000660
C                                                                       00000670
C     REPEAT FOR NEXT RECORD                                            00000680
C                                                                       00000690
      GO TO 100                                                         00000700
C                                                                       00000710
  600 CONTINUE                                                          00000720
C                                                                       00000730
      RETURN                                                            00000740
      END                                                               00000750
```

```
C     ******************************************************************
C
C     LSWT07(IOINIT)
C
C     OPENS DISK FILES FOR FOR I/O.  FILENAMES ARE PROMPTED FROM THE
C     CONSOLE.
C
C     ******************************************************************
C
      SUBROUTINE IOINIT
C
      CHARACTER*14 IPFILE,OPFILE,MSF1,MSF2,MSF3,MSF4
C
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
      DATA MSF1,      MSF2,      MSF3,      MSF4
     1  / 'MSF1.DAT','MSF2.DAT','MSF3.DAT','MSF4.DAT'/
C
C     ******************************************************************
C
C     ASSIGN DEFAULT VALUES TO I/O UNIT POINTERS
C
      IOU1  = 0
      IOU5  = 5
      IOU6  = 6
      IOU7  = IOU1
      IOU8  = 8
      IOU10 = 10
      IOU11 = 11
      IOU12 = 12
      IOU13 = 13
C
C     CLEAR THE DISPLAY
C
      CALL CLRSCN
C
C     PLACE CURSOR AT ROW 10 COLUMN 1
C
      CALL LOCATE(10,1)
C
C     PROMPT FOR I/O FILES AND READ USER RESPONSE
C
      WRITE(IOU1,10)
   10 FORMAT('ENTER INPUT FILENAME  ===> ' \)
      READ(IOU1,11) IPFILE
   11 FORMAT(A)
      WRITE(IOU1,12)
   12 FORMAT(' ENTER OUTPUT FILENAME ===> ' \)
      READ(IOU1,11) OPFILE
C
C     REDIRECT OUTPUT FILE IF REQUESTED
```

```
C
      IF(OPFILE.EQ.'CON') IOU6 = IOU1
C
C     OPEN STANDARD I/O FILES
C
C       IOU5  STANDARD INPUT
C       IOU6  STANDARD OUTPUT
C       IOU7  TRACE FILE ASSIGNED TO CONSOLE OR STANDARD OUTPUT
C       IOU8  TEMPORARY OUTPUT BUFFER FILE
C
                    OPEN(IOU5, FILE = IPFILE, STATUS = 'OLD')
      IF(IOU6.EQ.6) OPEN(IOU6, FILE = OPFILE, STATUS = 'NEW')
                    OPEN(IOU8, FILE = 'IOBUFFER.TMP', STATUS = 'NEW')
C
C     OPEN MASS STORAGE FILES
C
C       NAME      USE                 SIZE(ELEMENTS)
C       MSF1.DAT  SURFACE AIC'S       2*(100*100)*NBLADES
C       MSF2.DAT  WAKE AIC'S          2*(100*20 )*NBLADES
C       MSF3.DAT  COMBINED AIC ARRAY  2*(100*100)
C       MSF4.DAT  WAKE COORDINATES    (24 *133)*NBLADES
C
      OPEN(IOU10, FILE=MSF1, STATUS='NEW', ACCESS='DIRECT', RECL = 400 )
      OPEN(IOU11, FILE=MSF2, STATUS='NEW', ACCESS='DIRECT', RECL = 80  )
      OPEN(IOU12, FILE=MSF3, STATUS='NEW', ACCESS='DIRECT', RECL = 400 )
      OPEN(IOU13, FILE=MSF4, STATUS='NEW', ACCESS='DIRECT', RECL = 1596)
C
C     CLEAR THE DISPLAY SCREEN AND HOME THE CURSOR
C
      CALL CLRSCN
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT07(IOFNSH)
C
C     CLOSES FILES OPENED FOR DISK I/O
C
C     ******************************************************************
C
      SUBROUTINE IOFNSH
C
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
C     KEEP STANDARD I/O FILES
C
      CLOSE(IOU5, STATUS = 'KEEP')
      CLOSE(IOU6, STATUS = 'KEEP')
C
C     DELETE IOBUFFER AND MASS STORAGE FILES
C
      CLOSE(IOU8 , STATUS = 'DELETE')
      CLOSE(IOU10, STATUS = 'DELETE')
      CLOSE(IOU11, STATUS = 'DELETE')
      CLOSE(IOU12, STATUS = 'DELETE')
      CLOSE(IOU13, STATUS = 'DELETE')
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT07(TRACE)
C
C     PLACES A MARKER AND TIMING INFORMATION IN THE OUTPUT STREAM TO
C     IOU7
C
C     ******************************************************************
C
      SUBROUTINE TRACE(MARKER)
C
      CHARACTER*6 MARKER
C
      COMMON/ CIPRT  / IPRT1,  IPRT2,  IPRT3,  IPRT4,  IPRT5,  IPRT6,
     1                 IPRT7,  IPRT8
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
      WRITE(IOU7,10) MARKER
   10 FORMAT('0','TRACE: ',A6,/)
C
C     DISPLAY STEP TIME
C
      CALL TIMER(4)
C
C     DISPLAY TOTAL TIME
C
      CALL TIMER(2)
C
C     INITIALIZE STEP TIME
C
      CALL TIMER(3)
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT07(TIMER)
C
C     DISPLAYS PROGRAM TIMING INFORMATION TO IOU7.  SYSTEM DEPENDENT
C     SUBROUTINE SYSTIM IS CALLED TO READ THE SYSTEM CLOCK FOR THE
C     CURRENT TIME.
C
C          OPTION    ACTION
C            1       INITIALIZE START TIME IN H0,M0,S0,D0; NO DISPLAY
C            2       DISPLAY TOTAL TIME T-T0
C            3       INITIALIZE STEP TIME IN H1,M1,S1,D1; NO DISPLAY
C            4       DISPLAY STEP TIME, T-T1
C
C     ******************************************************************
C
      SUBROUTINE TIMER(OPTION)
C
      IMPLICIT INTEGER (A-Z)
C
      CHARACTER*49 FMT
      CHARACTER*1  FMTE(49)
      EQUIVALENCE  (FMT,FMTE)
C
      CHARACTER*10 TITLE(2)
C
      COMMON/ CTIMER / H0,     M0,     S0,     D0,     H1,     M1,
     1                 S1,     D1
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
      DATA FMT/'(('' '',A10,'' (H:M:S.D) '',IX,'':'',IX,'':'',IX,''.'',I2
     1.2)'/
C
      DATA TITLE(1)/'TOTAL TIME'/
      DATA TITLE(2)/'STEP TIME '/
C
C     ******************************************************************
C
      CALL SYSTIM(H,M,S,D)
C
      GO TO (100,200,300,400),OPTION
C
  100 CONTINUE
C
C     INITIALIZE START TIME
C
      H0 = H
      M0 = M
      S0 = S
      D0 = D
C
```

```
      GO TO 900
C
  200 CONTINUE
C
C
C     DISPLAY TOTAL TIME
C
      H = H-H0
      M = M-M0
      S = S-S0
      D = D-D0
C
      ITITLE = 1
      GO TO 500
C
  300 CONTINUE
C
C     INITIALIZE STEP TIME
C
      H1 = H
      M1 = M
      S1 = S
      D1 = D
C
      GO TO 900
C
  400 CONTINUE
C
C     DISPLAY STEP TIME
C
      H = H-H1
      M = M-M1
      S = S-S1
      D = D-D1
C
      ITITLE = 2
      GO TO 500
C
  500 CONTINUE
C
C     SCALE TIME COMPONENTS FOR DISPLAY
C
      IF(D.GE.0) GO TO 600
      D = D+100
      S = S-1
C
  600 CONTINUE
C
      IF(S.GE.0) GO TO 700
      S = S+60
      M = M-1
C
```

```
  700 CONTINUE
C
      IF(M.GE.0) GO TO 800
      M = M+60
      H = H-1
C
  800 CONTINUE
C
C     ADJUST FORMAT FOR SIZE OF TIME COMPONENTS
C
      FMTE(25) = '2'
      IF(H.LT.10) FMTE(25) = '1'
C
      FMTE(32) = '2'
      IF(M.LT.10) FMTE(32) = '1'
C
      FMTE(39) = '2'
      IF(S.LT.10) FMTE(39) = '1'
C
C     REPORT IT
C
      WRITE(IOU7,FMT) TITLE(ITITLE),H,M,S,D
C
  900 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT07(LOCATE)
C
C     POSITIONS THE CONSOLE DISPLAY CURSOR BY WRITING ANSI TERMINAL
C     CONTROL SEQUENCES TO UNIT IOU1.
C
C     ******************************************************************
C
      SUBROUTINE LOCATE(IROW,ICOL)
C
      CHARACTER*8 CUP
      CHARACTER*1 ECUP(8)
      EQUIVALENCE (CUP,ECUP(1))
C
      CHARACTER*10 FMT
      CHARACTER*1  EFMT(10)
      EQUIVALENCE  (FMT,EFMT(1))
C
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
      DATA FMT     / '(''  '',A? \)' /
C
C     ******************************************************************
C
C     ANSI CONTROL SEQUENCE: CUP = ESC['ROW';'COLUMN'H
C
      IR1 = IROW/10
      IR2 = IROW-IR1*10
      IC1 = ICOL/10
      IC2 = ICOL-IC1*10
C
      ECUP(1) = CHAR(27)
      ECUP(2) = CHAR(91)
      IPOS = 3
C
      IF(IR1.EQ.0) GO TO 100
      ECUP(IPOS) = CHAR(IR1+48)
      IPOS = IPOS + 1
  100 CONTINUE
C
      ECUP(IPOS) = CHAR(IR2+48)
      IPOS = IPOS + 1
C
      ECUP(IPOS) = CHAR(59)
      IPOS = IPOS + 1
C
      IF(IC1.EQ.0) GO TO 200
      ECUP(IPOS) = CHAR(IC1+48)
      IPOS = IPOS + 1
  200 CONTINUE
```

```
C
      ECUP(IPOS) = CHAR(IC2+48)
      IPOS = IPOS + 1
C
      ECUP(IPOS) = CHAR(72)
C
      EFMT(7) = CHAR(IPOS+48)
      WRITE (IOU1,FMT) CUP
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT07(CLRSCN)
C
C     CLEARS THE CONSOLE DISPLAY AND HOMES THE CURSOR BY WRITING ANSI
C     TERMINAL CONTROL SEQUENCES TO UNIT IOU1.
C
C     ******************************************************************
C
      SUBROUTINE CLRSCN
C
      CHARACTER*4  ED
      CHARACTER*1  EED(4)
      EQUIVALENCE  (ED,EED(1))
      CHARACTER*10 FMT
C
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
      DATA FMT     / '(\'' \'',A4 \)' /
C
C     ******************************************************************
C
C     ANSI CONTROL SEQUENCE: ED = ESC[2J
C
      EED(1) = CHAR(27)
      EED(2) = CHAR(91)
      EED(3) = CHAR(50)
      EED(4) = CHAR(74)
C
      WRITE (IOU1,FMT) ED
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(WNDPRO)
C
C     DETERMINES THE ONSET WIND PROFILE DUE TO THE ATMOSPHERIC BOUNDARY
C     LAYER AND TOWER SHADOW. IF ITRAN = 0 THE AXIS SYSTEM IS WIND AXIS.
C     OTHERWISE IT IS SHAFT AXIS AND THE INPUT COORDINATES ARE
C     TRANSFORMED TO WIND AXIS FOR DETERMINATION OF THE PROFILE.  THE
C     WIND VECTOR IS THEN RETURNED IN THE SAME AXIS SYSTEM AS INDICATED
C     BY 'ITRAN'.
C
C     ******************************************************************
C
      SUBROUTINE WNDPRO(ITRAN,X,Y,Z,VW1,VW2,VW3,AKZW)
C
      COMMON/ CONSTS / PI,      RPD
      COMMON/ CWIND  / ALPS1,   ALPS2,  HO,      VWO,     E,
     1                 ITSTYP,  TSTAB(100),      NXTS,    NYTS,
     2                 TO,      TP,      TPO,     TPS,     TK,
     3                 AB(5,2),          AT(5,2)
      COMMON/ IFLAGS / ISTDY,   IWFLAG, LSFLAG, IRFLAG
      COMMON/ PARMS  / RADIUS,  OMEGA,  OMEGAR, RHO,     AMU
C
C     ******************************************************************
C
      VW1  = 0.0
      VW2  = 0.0
      VW3  = VWO
      AKZW = VW3/OMEGAR
C
      IF(ISTDY.EQ.0) GO TO 200
      IF(ITRAN.EQ.0) GO TO 100
C
C     TRANSFORM (X,Y,Z) FROM SHAFT TO WIND AXIS
C
      CALL ROTATE(Z,X,ALPS2)
      CALL ROTATE(Y,Z,ALPS1)
C
  100 CONTINUE
C
C     HEIGHT ABOVE GROUND (POSITIVE Y POINTS DOWN)
C
      H = HO-Y
C
C     WIND PROFILE
C
      VW1 = 0.0
      VW2 = 0.0
      VW3 = VWO*(H/HO)**E
C
C     ADD TOWER SHADOW (TS IS VELOCITY DEFICIENCY FRACTION)
C
```

```
      GO TO (110,120,130), ITSTYP+1
C
  110 CONTINUE
C
C     NO TOWER SHADOW INCLUDED IN WIND PROFILE
C
      TS = 0.0
      GO TO 140
C
  120 CONTINUE
C
C     TOWER SHADOW EQUATION
C
      TS = 0.0
      PSI = ATAN2(-X,Y)
      IF( (PSI.LT.(TPS-TPO)) .OR. ((TPS+TPO).LT.PSI) ) GO TO 130
      TS = TO + TP * COS((PI*TK/TPO)*(PSI-TPS))
      GO TO 140
C
  130 CONTINUE
C
C     TOWER SHADOW TABLE
C
      CALL BVTINT(X,TS,Y,NXTS,NYTS,TSTAB,INOREX)
      IF(INOREX.NE.0) TS = 0.0
C
  140 CONTINUE
C
      VW3  = VW3 * (1.0-TS)
      AKZW = VW3 / OMEGAR
C
      IF(ITRAN.EQ.0) GO TO 200
C
C     RESOLVE WIND VECTOR TO SHAFT AXIS
C
      CALL ROTATE(VW2,VW3,-ALPS1)
      CALL ROTATE(VW3,VW1,-ALPS2)
C
  200 CONTINUE
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(FLPVEL)
C
C     CALCULATES THE ONSET VELOCITY DUE TO BLADE FLAPPING MOTION.  THE
C     FLAPPING VELOCITY IS RESOLVED IN SHAFT AXIS.
C
C     ******************************************************************
C
      SUBROUTINE FLPVEL(PSIT,X,Y,Z,VF1,VF2,VF3)
C
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
C
      DATA ZERO / 1.0E-06 /
C
C     ******************************************************************
C
C     DETERMINE AZIMUTH AND RADIUS OF COORDINATES
C
      PSIP = 0.0
      IF(ABS(X).GT.ZERO.AND.ABS(Y).GT.ZERO) PSIP = ATAN2(-X,Y)
      R = SQRT(X**2+Y**2+Z**2)
      R = R*COS(PSIP-PSIT)
C
C     FLAPPING ANGULAR DISPLACEMENT AND VELOCITY
C
      BETA  = AB(1,1)
      BETAD = 0.0
      NBH   = IFIX(AB(1,2)+0.1)
C
      DO 100   N = 1,NBH
      BETA  = BETA+AB(N+1,1)*COS(N*PSIT)+AB(N+1,2)*SIN(N*PSIT)
      BETAD = BETAD-AB(N+1,1)*SIN(N*PSIT)*N+AB(N+1,2)*COS(N*PSIT)*N
  100 CONTINUE
C
      BETAD = BETAD*OMEGA
C
C     VELOCITY COMPONENTS IN SHAFT AXIS AT (X,Y,Z)
C
      VF1 = 0.0
      VF2 = 0.0
      VF3 = R*BETAD
      CALL ROTATE(VF3,VF2,BETA)
      CALL ROTATE(VF1,VF2,PSIT)
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(NRMLBC)
C
C     CALCULATES THE NORMALWASH BOUNDARY CONDITION FROM SURFACE NORMAL
C     VECTORS AND ONSET VELOCITY
C
C     ******************************************************************
C
      SUBROUTINE NRMLBC(PSIT)
C
      REAL NRMWSH
C
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CVLCTY / VFS1(5,20),     VFS2(5,20),     VFS3(5,20),
     1                 VFS1M(20),      VFS2M(20),      VFS3M(20),
     2                 VI1(5,20),      VI2(5,20),      VI3(5,20),
     3                 VI1M(20),       VI2M(20),       VI3M(20),
     4                 VREL(20),       VIND(20),       QREL(20)
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ NMWSH  / NRMWSH(100)
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLCRD / XS(6,24),       YS(6,24),       ZS(6,24),
     1                 XCOL(5,20),     YCOL(5,20),     ZCOL(5,20)
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ PNLVEC / AN1(5,20),      AN2(5,20),      AN3(5,20),
     1                 AT1(5,20),      AT2(5,20),      AT3(5,20),
     2                 AQC1(20),       AQC2(20),       AQC3(20)
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
      DO 300   N = 1,NP
C
      IF(ISTDY.EQ.1) GO TO 100
C
      CALL WNDPRO(0,0.0,0.0,0.0,VW1,VW2,VW3,AKZW)
C
      VF1 = 0.0
      VF2 = 0.0
      VF3 = 0.0
C
  100 CONTINUE
C
      DO 300   M = 1,MP
C
```

```
      LP = (M-1)*NP+N
C
C     BUILD FREESTREAM VECTOR FROM WIND PROFILE,
C     FLAPPING, AND ROTATIONAL VELOCITIES
C
C
      IF(ISTDY.EQ.0) GO TO 200
C
      CALL WNDPRO(1,XCOL(M,N),YCOL(M,N),ZCOL(M,N),VW1,VW2,VW3,AKZW)
C
      CALL FLPVEL(PSIT,XCOL(M,N),YCOL(M,N),ZCOL(M,N),VF1,VF2,VF3)
C
  200 CONTINUE
C
      VFS1(M,N) = VW1+VF1+( OMEGAR*YCOL(M,N))
      VFS2(M,N) = VW2+VF2+(-OMEGAR*XCOL(M,N))
      VFS3(M,N) = VW3+VF3
C
C     NORMALWASH BOUNDARY CONDITION
C
      NRMWSH(LP) = -( AN1(M,N)*VFS1(M,N)
     1               +AN2(M,N)*VFS2(M,N)
     2               +AN3(M,N)*VFS3(M,N) ) / OMEGAR
C
  300 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(CIRPRM)
C
C     CALCULATES PARAMETERS DERIVED FROM THE DISTRIBUTED CIRCULATION
C     SOLUTION
C
C     ******************************************************************
C
      SUBROUTINE CIRPRM
C
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
C
C     ******************************************************************
C
      DO 100  J = 1,NP
C
      JP = (MP-1)*NP+J
C
C     BOUND (TRAILING EDGE) CIRCULATION
C
      BCIRC(J) = CIRC(JP)*TECF
C
C     LEADING EDGE SUCTION PARAMETER
C
      GO(J) = 1.0
      IF(MP.EQ.1) GO TO 100
      GXKM1 = CIRC(J)/BCIRC(J)/AKM(1)
      GXKM2 = CIRC(J+NP)/BCIRC(J)/AKM(2)
      GO(J) = (GXKM2-GXKM1)/(XKM(2)-XKM(1))*(-XKM(1))+GXKM1
C
  100 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(BEVEL)
C
C     CALCULATES THE BLADE ELEMENT VELOCITIES
C
C     ******************************************************************
C
      SUBROUTINE BEVEL(PSIT)
C
      DIMENSION DSWT(100)
C
      COMMON/ ALSLP  / ALS(20),        ALSNEW(20),     ALSOLD(20)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CDLS   / DLS(25)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CVLCTY / VFS1(5,20),     VFS2(5,20),     VFS3(5,20),
     1                 VFS1M(20),      VFS2M(20),      VFS3M(20),
     2                 VI1(5,20),      VI2(5,20),      VI3(5,20),
     3                 VI1M(20),       VI2M(20),       VI3M(20),
     4                 VREL(20),       VIND(20),       QREL(20)
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ PNLVEC / AN1(5,20),      AN2(5,20),      AN3(5,20),
     1                 AT1(5,20),      AT2(5,20),      AT3(5,20),
     2                 AQC1(20),       AQC2(20),       AQC3(20)
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
      DO 100   N = 1,NP
C
C
      VFS1M(N) = 0.0
      VFS2M(N) = 0.0
      VFS3M(N) = 0.0
C
      VI1M(N) = 0.0
      VI2M(N) = 0.0
      VI3M(N) = 0.0
```

```
C
      DO 200   M = 1,MP
C
C     NORMAL INDUCED VELOCITY
C
      VSIN = 0.0
C
      DO 300   I = 1,MP
      KP   = (I-1)*NP+N
      ISTR = (I-1)*MP+M
      VSIN = VSIN + DLS(ISTR)/CHDCGL(N)*CIRC(KP)
  300 CONTINUE
C
      VSIN = VSIN/RADIUS*2.0*PI/ALS(N)
      VFSN = VFS1(M,N)*AN1(M,N)
     1     + VFS2(M,N)*AN2(M,N)
     2     + VFS3(M,N)*AN3(M,N)
      VN   = -VFSN-VSIN
C
C     TANGENTIAL INDUCED VELOCITY
C
      LP = (M-1)*NP+N
      READ(IOU12, REC = NPANEL+LP) (DSWT(KP), KP = 1,NPANEL)
C
      VT = 0.0
C
      DO 400   KP = 1,NPANEL
      VT = VT+DSWT(KP)*CIRC(KP)
  400 CONTINUE
C
      VT = VT/RADIUS
C
C     RESOLVE TO COMPONENTS IN FIXED REFERENCE SYSTEM
C
      VI1(M,N) = VN*AN1(M,N) + VT*AT1(M,N)
      VI2(M,N) = VN*AN2(M,N) + VT*AT2(M,N)
      VI3(M,N) = VN*AN3(M,N) + VT*AT3(M,N)
C
C     TRANSFORM VELOCITIES TO BLADE ELEMENT COORDINATE SYSTEM
C
      ANGLE = -PSIT
      CALL ROTATE(VI1(M,N),VI2(M,N),ANGLE)
      CALL ROTATE(VFS1(M,N),VFS2(M,N),ANGLE)
      ANGLE = 0.0
      DY    = -AN2(M,N)
      DX    =  AN3(M,N)
      IF(ABS(DY).GT.1.0E-06) ANGLE = -ATAN2(DY,DX)
      CALL ROTATE(VI2(M,N),VI3(M,N),ANGLE)
      CALL ROTATE(VFS2(M,N),VFS3(M,N),ANGLE)
C
  200 CONTINUE
C
```

```
C     GET CHORDWISE AVERAGE
C
      CALL CWAVG ( VFS1(1,N), VFS1M(N) )
      CALL CWAVG ( VFS2(1,N), VFS2M(N) )
      CALL CWAVG ( VFS3(1,N), VFS3M(N) )
C
      CALL CWAVG ( VI1(1,N), VI1M(N) )
      CALL CWAVG ( VI2(1,N), VI2M(N) )
      CALL CWAVG ( VI3(1,N), VI3M(N) )
C
C     ELEMENT INDUCED VELOCITY
C
      VIND(N) = VI1M(N)
C
C     ELEMENT RELATIVE VELOCITY AND DYNAMIC PRESSURE
C
      VREL1   = VFS1M(N) + VI1M(N)
      VREL2   = VFS2M(N) + VI2M(N)
      VREL3   = VFS3M(N) + VI3M(N)
      VREL(N) = SQRT( VREL1**2 + VREL2**2 + VREL3**2 )
      QREL(N) = 0.5*RHO*VREL(N)**2
C
C     COSINE OF AERODYNAMIC SWEEP ANGLE FROM VECTOR PRODUCT OF
C     VELOCITY AND CIRCULATION (CIRCULATION ASSUMED TO BE ALONG
C     QUARTER CHORD FOR THIS CALCULATION) -- NOTE THAT COSINE OF SWEEP
C     IS THE SINE OF INCLUDED ANGLE
C
      CALL VPANG(AQC1(N),AQC2(N),AQC3(N),VREL1,VREL2,VREL3,CLAMQC(N))
C
  100 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(CWAVG)
C
C     CALCULATES THE CHORDWISE AVERAGE 'YM' OF THE 'MP' POINTS PASSED
C     THROUGH THE DUMMY ARRAY 'Y'
C
C     ******************************************************************
C
      SUBROUTINE CWAVG(Y,YM)
C
      COMMON/ CGRID  / NGRID(4),       XG(6),  YG(24), XCG(5)
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
C
      DIMENSION Y(1)
C
C     ******************************************************************
C
      YM = Y(1)
      IF(MP.EQ.1) GO TO 200
C
      YLE = (Y(2)-Y(1))/(XCG(1)-XCG(2))*(-XCG(1))+Y(1)
      YTE = (Y(MP)-Y(MP-1))/(XCG(MP)-XCG(MP-1))*(1.0-XCG(MP))+Y(MP)
      YM  = (YLE+Y(1))/2.0*XCG(1) + (Y(MP)+YTE)/2.0*(1.0-XCG(MP))
C
      DO 100  I=1,MP-1
      YM = YM + (Y(I+1)+Y(I))/2.0*(XCG(I+1)-XCG(I))
  100 CONTINUE
C
  200 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(BEALPH)
C
C     CALCULATES THE BLADE ELEMENT ANGLES OF ATTACK
C
C     ******************************************************************
C
      SUBROUTINE BEALPH
C
      COMMON/ ALSLP  / ALS(20),        ALSNEW(20),     ALSOLD(20)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),
     2                 TWSTCG(20)
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CONTRL / TOL1,   TOL2,   TOL3,   ICYL1,  ICYL2,  ICYL3,
     1                 ICYL4
      COMMON/ CVLCTY / VFS1(5,20),     VFS2(5,20),     VFS3(5,20),
     1                 VFS1M(20),      VFS2M(20),      VFS3M(20),
     2                 VI1(5,20),      VI2(5,20),      VI3(5,20),
     3                 VI1M(20),       VI2M(20),       VI3M(20),
     4                 VREL(20),       VIND(20),       QREL(20)
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
C
      DATA ZERO /1.0E-06/
C
C     ******************************************************************
C
C     RESET LIFT SLOPE CONVERGENCE FLAG
C
      LSFLAG = 0
C
      DO 100   J = 1,NP
C
      JP = (MP-1)*NP+J
C
C     ONSET VELOCITY ANGLE OF ATTACK
C
```

```
      ALPHAW(J) = 0.0
      IF(ABS(VFS3M(J)).GT.ZERO) ALPHAW(J) = ATAN2(VFS3M(J),VFS1M(J))
C
C     EFFECTIVE ANGLE OF ATTACK
C
      RE(J)     = RHO*VREL(J)*CHDCGL(J)*RADIUS/AMU
      CLO       = 2.0*BCIRC(J)/VREL(J)/CHDCGL(J)/RADIUS
      CDS(J)    = -(CLO**2/(2.0*PI))/CLAMLE(J) * (GO(J)**2-1.0)
      ALPHAE(J) = ASIN(CLO/ALS(J))/CLAMQC(J)
C
C     ELEMENT AIRFOIL CHARACTERISTICS WITH SWEEP CORRECTIONS
C
      ALTAB = (ALPHAE(J)+AZLLC(J))/RPD
      CALL AFCOEF(J,ALTAB,RE(J),CLT(J),CDT(J),CMT(J),INOREX)
C
      CLT(J) = CLT(J)*CLAMQC(J)**2
C
C     'CDF' IS PORTION OF PROFILE DRAG COEFFICIENT ATTRIBUTABLE TO
C     FRICTION.  CDT-CDF IS THEN PRESSURE DRAG AND MUST BE CORRECTED
C     FOR SWEEP
C
      CDF    = 0.006
      CDP    = CDT(J) - CDF
      CDT(J) = CDF + CDP*CLAMQC(J)**3
C
C     RESET SUCTION FORCE INCREMENT IF STALLED
C
      IF(INOREX.NE.0) CDS(J) = 0.0
C
C     CALCULATE NEW LIFT CURVE SLOPE AND CHECK CONVERGENCE
C
      ALSNEW(J) = CLT(J)/SIN(ALPHAE(J))
C
      TOLMAX = ABS(1.0/ALSNEW(J)**2)
      IF(TOLMAX.LT.0.01) TOLMAX = 0.01
      IF(TOL2.GT.0.0) TOLMAX = TOL2
      TOL = ABS((ALSNEW(J)-ALS(J))/ALSNEW(J))
      IF(TOL.GT.TOLMAX) LSFLAG = 1
C
      ALSOLD(J) = ALS(J)
      ALS(J)    = ALSNEW(J)
C
C     INFLOW 'N' AND INDUCED 'I' ANGLES OF ATTACK
C       USE ANGLE RESOLUTION BELOW STALL, VELOCITY RESOLUTION ABOVE
C       STALL
C
      IF(INOREX.NE.0) GOTO 110
      ALPHAN(J) = ALPHAE(J)-ALPHAG(J)+AZLLC(J)
      ALPHAI(J) = ALPHAN(J)-ALPHAW(J)
      GOTO 100
  110 CONTINUE
      ALPHAN(J) = ATAN2(VFS3M(J)+VI3M(J),VFS1M(J)+VI1M(J))
```

```
      ALPHAI(J) = ATAN2(VI3M(J),VFS1M(J)+VI1M(J))
C
  100 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(AFCOEF)
C
C     CALLS TABLE LOOKUPS, ADDS ANALYTICAL EXTENSIONS,  AND PERFORMS
C     AVERAGING TO GET CL,CD, AND CM AT BLADE ELEMENT 'J'.
C
C     ******************************************************************
C
      SUBROUTINE AFCOEF(J,ALTAB,REN,CL,CD,CM,INOREX)
C
      COMMON/ AFTABL / NAFT,   IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ CCIRC  / CIRC(100),      DCIRC(100),     ACIRC(100),
     1                 CIRCR(100),     BCIRC(20),      GO(20),
     2                 TECF,           AKM(6),         XKM(5)
      COMMON/ CONSTS / PI,     RPD
C
C     ******************************************************************
C
      JTAB1 = JTAB(J,1)
      JTAB2 = JTAB(J,2)
      IPASS = 1
      JTABP = JTAB1
C
      NRCLP = NRCL(JTABP)
      NACLP = NACL(JTABP)
      NRCDP = NRCD(JTABP)
      NACDP = NACD(JTABP)
      NRCMP = NRCM(JTABP)
      NACMP = NACM(JTABP)
C
  100 CONTINUE
C
      CALL BVTINT(REN,CLP,ALTAB,NRCLP,NACLP,CLTAB(1,JTABP),INEXCL)
      CALL BVTINT(REN,CDP,ALTAB,NRCDP,NACDP,CDTAB(1,JTABP),INEXCD)
      CALL BVTINT(REN,CMP,ALTAB,NRCMP,NACMP,CMTAB(1,JTABP),INEXCM)
C
C     IF OUT OF TABLE RANGE THEN USE ANALYTICAL MODEL IF REQUESTED,
C     OTHERWISE KEEP TABLE EXTRAPOLATIONS
C
      IF(INEXCL.EQ.0.AND.INEXCD.EQ.0.AND.INEXCM.EQ.0) GO TO 110
```

```
      IF(IAFTEX.EQ.0) GO TO 110
C
C     GET CL,CD, AND CM AT STALL (ASSUMED TO BE SAME AS TABLE LIMIT)
C
      ITABL = NRCLP+NACLP
      ITABD = NRCDP+NACDP
      ITABM = NRCMP+NACMP
C
      ALPSL = CLTAB(ITABL,JTABP)
      ALPSD = CDTAB(ITABD,JTABP)
      ALPSM = CMTAB(ITABM,JTABP)
C
      CALL BVTINT(REN,CLS,ALPSL,NRCLP,NACLP,CLTAB(1,JTABP),INOREX)
      CALL BVTINT(REN,CDS,ALPSD,NRCDP,NACDP,CDTAB(1,JTABP),INOREX)
      CALL BVTINT(REN,CMS,ALPSM,NRCMP,NACMP,CMTAB(1,JTABP),INOREX)
C
C     EXTEND TABLE ANALYTICALLY
C
      CALL AFTEXT(JTABP,ALTAB,ALPSL,ALPSD,CLS,CDS,CLEX,CDEX)
C
      IF(INEXCL.NE.0) CLP = CLEX
      IF(INEXCD.NE.0) CDP = CDEX
      IF(INEXCM.NE.0) CMP = CMS
      INOREX = 0
      IF(INEXCL.NE.0) INOREX = 1
C
      IF(IPASS.EQ.2) GO TO 120
C
  110 CONTINUE
C
      CL = CLP
      CD = CDP
      CM = CMP
C
C     REPEAT FOR SECOND TABLE IF AVERAGING USED AT THIS STATION
C
      IF(JTAB1.EQ.JTAB2) GO TO 120
C
      IPASS = 2
      JTABP = JTAB2
C
      NRCLP = NRCL(JTABP)
      NACLP = NACL(JTABP)
      NRCDP = NRCD(JTABP)
      NACDP = NACD(JTABP)
      NRCMP = NRCM(JTABP)
      NACMP = NACM(JTABP)
C
      GO TO 100
C
  120 CONTINUE
C
```

```
      CL = CLP*AFTWF(J)+CL*(1.0-AFTWF(J))                               00001050
      CD = CDP*AFTWF(J)+CD*(1.0-AFTWF(J))                               00001060
      CM = CMP*AFTWF(J)+CM*(1.0-AFTWF(J))                               00001070
C                                                                       00001080
      RETURN                                                            00001090
      END                                                               00001100
```

```
C     ******************************************************************
C
C     LSWT08(AFTEXT)
C
C     EXTENDS AIRFOIL TABLES WITH ANALYTICAL METHODS.
C
C     ******************************************************************
C
      SUBROUTINE AFTEXT(JTAB,ALTAB,ALPSL,ALPSD,CLS,CDS,CL,CD)
C
      COMMON/ AFTABL / NAFT,   IDAFT(4),       AFTOC(4),       IAFTEX,
     1                 STP1(4),        STP2(4),        STP3(4),
     2                 STP4(4),        STP5(4),        STP6(4),
     3                 CLTAB(200,4),   NRCL(4),        NACL(4),
     4                 CDTAB(200,4),   NRCD(4),        NACD(4),
     5                 CMTAB(100,4),   NRCM(4),        NACM(4)
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),
     2                 TWSTCG(20)
      COMMON/ CONSTS / PI,     RPD
C
C     ******************************************************************
C
      ALP    = ALTAB * RPD
      ALPSL  = ALPSL * RPD
      ALPSD  = ALPSD * RPD
      ABSALP = ABS(ALP)
C
      SA   = SIN(ALP)
      CA   = COS(ALP)
      SASL = SIN(ALPSL)
      SASD = SIN(ALPSD)
      CASL = COS(ALPSL)
      CASD = COS(ALPSD)
C
      IF(IAFTEX.GT.1) GO TO 200
C
C     ------------------------------------------------------------
C     IAFTEX = 1  AEROVIRONMENT METHOD
C     FLAT PLATE NORMAL FORCE COEFFICIENT WITH MODIFICATIONS FOR
C     EFFECTIVE ASPECT RATIO BASED ON PRANDTL TIP LOSS FACTOR
C     ------------------------------------------------------------
C
C     APPROXIMATE 'PRANDTL LOSS FUNCTION' 3D/2D ALPHA
C
      F = ABS(ALPHAE(J)/(ALPHAW(J)+ALPHAG(J)-AZLLC(J)))
      CN = 1.98
      IF(F.GE.0.99999) GO TO 100
      AR = 2.0
      IF(F.GT.0.5) AR = 2.0 * F/(1-F)
      CN = 1.98 - 0.81 * TANH(12.22/AR)
  100 CN=AMIN1( 1.0 / ( 0.222 + 0.283 / SIN(ABSALP) ),CN )
```

```
      IF(ALP.LT.0.0) CN = -CN
C
C     LIFT AND DRAG COEFFICIENTS
C
      CL = CN * CA
      CD = CN * SA
C
      GO TO 1000
C
  200 CONTINUE
      IF(IAFTEX.GT.2) GO TO 300
C
C     ----------------------------------------------
C     IAFTEX = 2  VITERNA'S METHOD
C     ----------------------------------------------
C
      AR    = STP1(JTAB)
      CDMAX = 1.11+0.018*AR
C
      B1 = CDMAX
      B2 = (CDS-B1*SASD**2)/CASD
C
      A1 = B1/2.0
      A2 = (CLS-CDMAX*SASL*CASL)*SASL/CASL**2
C
C     LIFT AND DRAG COEFFICIENTS
C
      CL = A1*2.0*SA*CA + A2*CA**2/SA
      CD = B1*SA**2 + B2*CA
C
      GO TO 1000
C
  300 CONTINUE
      IF(IAFTEX.GT.3) GO TO 1000
C
C     ----------------------------------------------
C     IAFTEX = 3  CMA METHOD
C     ----------------------------------------------
C
      ALPR = STP1(JTAB)*RPD
      CL45 = STP2(JTAB)
      CD45 = STP3(JTAB)
      CL90 = STP4(JTAB)
      CD90 = STP5(JTAB)
      BPR  = STP6(JTAB)
C
      IF(ALP.GT.ALPR) GOTO 310
C
C     REGION 2 -- TRANSITION FROM STALL TO RECOVERY
C                 ALPS < ALP < ALPR
C
      SAR = SIN(ALPR)
```

```
      CAR = COS(ALPR)
      CLR = (2.0*CL45*SAR*CAR)
      CDR = (2.0*CD45*SAR**2 )
C
      A1 = (CLR*CASL**2*SAR-CLS*CAR**2*SASL) /
     1     (2.0*CASL*CAR*(SAR**2*CASL-SASL**2*CAR))
      A2 = (CLS-2.0*A1*SASL*CASL)*SASL/CASL**2
      B1 = (CDS*CAR-CDR*CASD)/(SASD**3*CAR-SAR**3*CASD)
      B2 = (CDR-B1*SAR**3)/CAR
      CL = A1*2.0*SA*CA + A2*CA**2/SA
      CD = B1*SA**3     + B2*CA
C
      GOTO 1000
C
  310 CONTINUE
      IF(ALP.GT.PI/4.0) GOTO 320
C
C     REGION 3 -- RECOVERY TO 45 DEG
C
      A1 = 2.0*CL45*BPR
      A2 = (CLR-2.0*CL45*BPR*SAR*CAR)/(1.0-2.0*SAR**2)
      B1 = (CDR-SQRT(2.0)*CD45*BPR*CAR)/(SAR**2-SQRT(2.0)/2.0*CAR)
      B2 = (CD45*BPR-B1/2.0)*SQRT(2.0)
      CL = A1*SA*CA+A2*(1.0-2.0*SA**2)
      CD = B1*SA**2+B2*CA
C
      GOTO 1000
C
  320 CONTINUE
C
C     REGION 4 -- 45 DEG TO 90 DEG
C
      CL = (2.0*CL45*SA*CA - CL90*(1-2.0*SA**2))       * BPR
      CD = (CD90*SA**2 + (2.0*CD45-CD90)/SQRT(2.0)*CA) * BPR
C
 1000 CONTINUE
C
      RETURN
      END
```

```
C     ******************************************************************
C
C     LSWT08(AEROPF)
C
C     CALCULATES ELEMENT, BLADE, AND ROTOR PERFORMANCE
C
C     ******************************************************************
C
      SUBROUTINE AEROPF(ITBLD,IPSIR,ISCYL)
C
      COMMON/ BELOAD / DFX(20),        DFXI(20),       DFXP(20),
     1                 DFZ(20),        DFZI(20),       DFZP(20),
     2                 DQ(20),         DQI(20),        DQP(20),
     3                 DPM(20)
      COMMON/ BLDELE / XCLE(20),       YCG(20),        RBE(20),
     1                 DBES(20),       CLAMLE(20),     CLAMQC(20),
     2                 CHDCGL(20),     BETAL(20),      AFTWF(20),
     3                 JTAB(20,2),     TOC(20),        RE(20),
     4                 CLT(20),        CDT(20),        CMT(20),
     5                 CDS(20)
      COMMON/ BLLOAD / CTRB,   CQRB,   CTWB,   CPWB,   BTHRST, BTORQ,
     1                 BTORQI, BTORQP, BPWR,   BPWRP,  BPWRI,  BPMOMT
      COMMON/ CALPHA / ALPHAG(20),     ALPHAE(20),     ALPHAW(20),
     1                 ALPHAN(20),     ALPHAI(20),     AZLLC(20),
     2                 TWSTCG(20)
      COMMON/ CONSTS / PI,     RPD
      COMMON/ CTCON  / CTRIN,  CTROUT, CTRUPD
      COMMON/ CVLCTY / VFS1(5,20),     VFS2(5,20),     VFS3(5,20),
     1                 VFS1M(20),      VFS2M(20),      VFS3M(20),
     2                 VI1(5,20),      VI2(5,20),      VI3(5,20),
     3                 VI1M(20),       VI2M(20),       VI3M(20),
     4                 VREL(20),       VIND(20),       QREL(20)
      COMMON/ CWIND  / ALPS1,  ALPS2,  HO,     VWO,    E,
     1                 ITSTYP, TSTAB(100),     NXTS,   NYTS,
     2                 TO,     TP,     TPO,    TPS,    TK,
     3                 AB(5,2),        AT(5,2)
      COMMON/ IFLAGS / ISTDY,  IWFLAG, LSFLAG, IRFLAG
      COMMON/ PARMS  / RADIUS, OMEGA,  OMEGAR, RHO,    AMU
      COMMON/ PNLNDX / MP,             NP,             NPANEL,
     1                 MP1,            NP1,            NPNLSQ,
     2                 NSGMNT,         JGLIM(4,2),     JCGLIM(4,2),
     3                 NPS(4),         NBLDS
      COMMON/ RTLOAD / CTRR,   CQRR,   CTWR,   CPWR,   RTHRST, RTORQ,
     1                 RTORQP, RTORQI, RPWR,   RPWRP,  RPWRI,  CTR,
     2                 CQR,    CTW,    CPW,    THRST,  TORQ,   TORQP,
     3                 TORQI,  PWR,    PWRP,   PWRI,   RPMOMT, PMOMT
      COMMON/ UNITS  / IOU1,   IOU5,   IOU6,   IOU7,   IOU8,
     1                 IOU10,  IOU11,  IOU12,  IOU13
C
C     ******************************************************************
C
      IF(ISTDY.EQ.0) GO TO 20
```

```
      IF(ISCYL.EQ.1) GO TO 20
C
C     RESET IF ENTRY FOR SAME BLADE AS PREVIOUS ENTRY
C
      IF(ITBLD.NE.NBLDS) GO TO 10
C
      IM1 = IPSIR - 1
      IF(IM1.EQ.0) IM1 = 1
C
      THRST = ( THRST * IPSIR - RTHRST ) / IM1
      TORQ  = ( TORQ  * IPSIR - RTORQ  ) / IM1
      TORQP = ( TOPQP * IPSIR - RTORQP ) / IM1
      TORQI = ( TORQI * IPSIR - RTORQI ) / IM1
C
      PWR   = ( PWR  * IPSIR - RPWR  ) / IM1
      PWRP  = ( PWRP * IPSIR - RPWRP ) / IM1
      PWRI  = ( PWRI * IPSIR - RPWRI ) / IM1
C
      PMOMT = ( PMOMT * IPSIR - RPMOMT ) / IM1
C
      CTR   = ( CTR * IPSIR - CTRR ) / IM1
      CQR   = ( CQR * IPSIR - CQRR ) / IM1
      CTW   = ( CTW * IPSIR - CTWR ) / IM1
      CPW   = ( CPW * IPSIR - CPWR ) / IM1
C
   10 CONTINUE
C
      RTHRST = RTHRST - BTHRST
      RTORQ  = RTORQ  - BTORQ
      RTORQP = RTORQP - BTORQP
      RTORQI = RTORQI - BTORQI
C
      RPWR  = RPWR  - BPWR
      RPWRP = RPWRP - BPWRP
      RPWRI = RPWRI - BPWRI
C
      RPMOMT = RPMOMT - BPMOMT
C
      CTRR = CTRR - CTRB
      CQRR = CQRR - CQRB
      CTWR = CTWR - CTWB
      CPWR = CPWR - CPWB
C
   20 CONTINUE
C
C     INITIALIZE BLADE PERFORMANCE FOR SUMMING
C
      BTHRST = 0.0
      BTORQ  = 0.0
      BTORQP = 0.0
      BTORQI = 0.0
      BPMOMT = 0.0
```

```
C
C     ------------------------------
C     LOOP 100 SWEEPS BLADE ELEMENTS
C     ------------------------------
C
      DO 100   J = 1,NP
C
      QS      = QREL(J)*CHDCGL(J)*DBES(J)*RADIUS**2
      QSC     = QS*CHDCGL(J)*RADIUS
C
C     ELEMENT FORCES (IN ELEMENT AXIS SYSTEM)
C
      DFXI(J) = (-CLT(J) * SIN(ALPHAN(J))
     1        +   CDS(J) * COS(ALPHAG(J)-AZLLC(J))) * QS
      DFXP(J) =   CDT(J) * COS(ALPHAN(J)) * QS
      DFX(J)  = DFXP(J) + DFXI(J)
C
      DFZI(J) = (CLT(J) * COS(ALPHAN(J))
     1        -  CDS(J) * SIN(ALPHAG(J)-AZLLC(J))) * QS
      DFZI(J) = DFZI(J) * COS(BETAL(J))
      DFZP(J) =  CDT(J) * SIN(ALPHAN(J)) * QS
      DFZP(J) = DFZP(J) * COS(BETAL(J))
      DFZ(J)  = DFZP(J) + DFZI(J)
C
C     PITCHING MOMENT DUE TO 2D AIRFOIL COEFFICIENT
C
      DPM(J)  =  CMT(J) * QSC
C
C     ELEMENT TORQUE - 'RBE' IS RADIUS OF ELEMENT WITH CONING AND STEADY
C     FLAPPING, THUS TORQUE IS TIP PATH PLANE TORQUE.  ACTUAL SHAFT
C     TORQUE DEPENDS ON TYPE OF JOINT BETWEEN ROTOR AND SHAFT.
C
      DQP(J) = DFXP(J) * RBE(J) * RADIUS
      DQI(J) = DFXI(J) * RBE(J) * RADIUS
      DQ(J)  = DQP(J)  + DQI(J)
C
C     BLADE FORCES AND TORQUES
C
      BTHRST = BTHRST + DFZ(J)
      BTORQ  = BTORQ  + DQ(J)
      BTORQP = BTORQP + DQP(J)
      BTORQI = BTORQI + DQI(J)
      BPMOMT = BPMOMT + DPM(J)
C
C     CHANGE ELEMENT FORCES AND TORQUES TO LOADINGS
C
      FAC     = DBES(J) / RADIUS
      DFX(J)  = DFX(J)  / FAC
      DFXI(J) = DFXI(J) / FAC
      DFXP(J) = DFXP(J) / FAC
      DFZ(J)  = DFZ(J)  / FAC
      DFZI(J) = DFZI(J) / FAC
```

```
      DFZP(J) = DFZP(J) / FAC
      DQP(J)  = DQP(J)  / FAC
      DQI(J)  = DQI(J)  / FAC
      DQ(J)   = DQ(J)   / FAC
      DPM(J)  = DPM(J)  / FAC
C
  100 CONTINUE
C
C     ---------------------------------
C     BLADE FORCES, TORQUES, AND POWERS
C     ---------------------------------
C
C     BLADE COEFFICIENTS
C
      CTRB = BTHRST / (RHO     * PI * RADIUS**2 * OMEGAR**2)
      CQRB = BTORQ  / (RHO     * PI * RADIUS**3 * OMEGAR**2)
      CTWB = BTHRST / (RHO/2.0 * PI * RADIUS**2 * VWO**2   )
      CPWB = BTORQ  / (RHO/2.0 * PI * RADIUS**2 * VWO**3   ) * OMEGA
C
C     BLADE POWER IN KW
C
      FAC   = OMEGA  / 737.562
      BPWR  = BTORQ  * FAC
      BPWRP = BTORQP * FAC
      BPWRI = BTORQI * FAC
C
C     ---------------------------------
C     ROTOR FORCES, TORQUES, AND POWERS
C     ---------------------------------
C
      IF(ITBLD.GT.1) GO TO 200
C
C     INITIALIZE ROTOR TOTALS FOR THIS AZIMUTH AT ENTRY FOR
C     FIRST BLADE
C
      RTHRST = 0.0
      RTORQ  = 0.0
      RTORQP = 0.0
      RTORQI = 0.0
C
      RPWR  = 0.0
      RPWRP = 0.0
      RPWRI = 0.0
C
      RPMOMT = 0.0
C
      CTRR = 0.0
      CQRR = 0.0
      CTWR = 0.0
      CPWR = 0.0
C
      IF(IPSIR.GT.1) GO TO 200
```

```
C
C     IF FIRST AZIMUTH THEN INITIALIZE DISK AVERAGES
C
      THRST = 0.0
      TORQ  = 0.0
      TORQP = 0.0
      TORQI = 0.0
C
      PMOMT = 0.0
C
      PWR  = 0.0
      PWRP = 0.0
      PWRI = 0.0
C
      CTR = 0.0
      CQR = 0.0
      CTW = 0.0
      CPW = 0.0
C
  200 CONTINUE
C
C     SUM THIS BLADES CONTIBUTION TO TOTALS FOR THIS AZIMUTH
C
      RTHRST = RTHRST + BTHRST
      RTORQ  = RTORQ  + BTORQ
      RTORQP = RTORQP + BTORQP
      RTORQI = RTORQI + BTORQI
C
      RPWR  = RPWR  + BPWR
      RPWRP = RPWRP + BPWRP
      RPWRI = RPWRI + BPWRI
C
      RPMOMT = RPMOMT + BPMOMT
C
      CTRR = CTRR + CTRB
      CQRR = CQRR + CQRB
      CTWR = CTWR + CTWB
      CPWR = CPWR + CPWB
C
      IF(ISTDY.NE.0) GO TO 300
C
C     DISK TOTALS FOR STEADY CASE
C
      RTHRST = RTHRST * NBLDS
      RTORQ  = RTORQ  * NBLDS
      RTORQP = RTORQP * NBLDS
      RTORQI = RTORQI * NBLDS
C
      RPWR  = RPWR  * NBLDS
      RPWRP = RPWRP * NBLDS
      RPWRI = RPWRI * NBLDS
C
```

```
      PMOMT = PMOMT * NBLDS
C
      CTRR = CTRR * NBLDS
      CQRR = CQRR * NBLDS
      CTWR = CTWR * NBLDS
      CPWR = CPWR * NBLDS
C
      GO TO 400
C
  300 CONTINUE
C
C     IF TOTALS AVAILABLE FOR THIS AZIMUTH THEN UPDATE
C     DISK AVERAGES
C
      IF(ITBLD.NE.NBLDS) GO TO 400
C
      IM1 = IPSIR - 1
C
      THRST = ( THRST * IM1 + RTHRST ) / IPSIR
      TORQ  = ( TORQ  * IM1 + RTORQ  ) / IPSIR
      TORQP = ( TOPQP * IM1 + RTORQP ) / IPSIR
      TORQI = ( TORQI * IM1 + RTORQI ) / IPSIR
C
      PWR   = ( PWR  * IM1 + RPWR  ) / IPSIR
      PWRP  = ( PWRP * IM1 + RPWRP ) / IPSIR
      PWRI  = ( PWRI * IM1 + RPWRI ) / IPSIR
C
      PMOMT = ( PMOMT * IM1 + RPMOMT ) / IPSIR
C
      CTR   = ( CTR * IM1 + CTRR ) / IPSIR
      CQR   = ( CQR * IM1 + CQRR ) / IPSIR
      CTW   = ( CTW * IM1 + CTWR ) / IPSIR
      CPW   = ( CPW * IM1 + CPWR ) / IPSIR

  400 CONTINUE
C
C     UPDATE FOR WAKE PARAMETER CALCULATION
C
      IF(ITBLD.EQ.NBLDS) CTROUT = CTR
      IF(ISTDY.EQ.0) CTROUT = CTRR
C
      RETURN
      END
```

```
;       ****************************************************************
;       SUBROUTINE SYSTIM
;
;       ASSEMBLER INTERFACE TO SYSTEM CLOCK FOR MS FORTRAN77, VER.3.2
;       UNDER MS-DOS 2.X
;
;       READS THE SYSTEM CLOCK AND RETURNS HOURS,
;       MINUTES, SECONDS, AND HUNDRETHS OF SECONDS
;
;       USAGE: CALL SYSTIM(H,M,S,D)
;
;       ARGUMENTS ARE ALL INTEGER*4
;       H = HOUR            RANGE 0-59
;       M = MINUTE                0-59
;       S = SECOND                0-59
;       D = 1/100 SECOND          0-99
;
;       ****************************************************************
;
        FRAME   STRUC
        SAVEBP  DW      ?
        SAVERET DD      ?
        D       DD      ?
        S       DD      ?
        M       DD      ?
        H       DD      ?
        FRAME   ENDS
;
        DATA    SEGMENT PUBLIC 'DATA'
        DATA    ENDS
;
        CODE    SEGMENT 'CODE'
        DGROUP  GROUP   DATA
                ASSUME  CS:CODE,DS:DGROUP,ES:DGROUP,SS:DGROUP
;
SYSTIM  PROC    FAR
        PUBLIC  SYSTIM
;
        PUSH    BP          ;SAVE FRAMEPOINTER ON STACK
        MOV     BP,SP
;
        MOV     AH,2CH      ;DOS SERVICE TO GET TIME
        INT     21H
;
        LES     BX,[BP]+H   ;HOURS
        MOV     AH,0
        MOV     AL,CH
        MOV     ES:[BX],AX
        MOV     WORD PTR ES:[BX+2],0
;
        LES     BX,[BP]+M   ;MINUTES
        MOV     AL,CL
```

```
        MOV     ES:[BX],AX
        MOV     WORD PTR ES:[BX+2],0
;
        LES     BX,[BP]+S   ;SECONDS
        MOV     AL,DH
        MOV     ES:[BX],AX
        MOV     WORD PTR ES:[BX+2],0
;
        LES     BX,[BP]+D   ;HUNDRETHS
        MOV     AL,DL
        MOV     ES:[BX],AX
        MOV     WORD PTR ES:[BX+2],0
;
        MOV     SP,BP
        POP     BP          ;RESTORE THE FRAME POINTER
        RET     16
;
SYSTIM  ENDP
;
CODE    ENDS
;
END
```

APPENDIX D

SAMPLE INPUT AND OUTPUT FILE LISTINGS

RECORD IMAGE OF INPUT FILE

```
MOD-0 (AILERON) *** CMA STALL MODEL *** 05
3.0       0.0       2.0       0.0       2.0       2.0       0.0
1.0
0.0       0.0       0.0
0.0       0.0       0.0       0.0
0.0       20.60     1.0       1.0
10.0      1.0       1.0       1.0
0.0       1.0       1.0       4000.0
2.0       2.0       4.0       63.9592
2.0       12.0      0.0       180.0     30.0      1.0
0.0       0.0       1.0       16.0      17.0      19.0
3.0       3.0       0.0       3.3333    13.3332
-.9170    0.0       0.0       3.0       0.0       0.0
2.0
3.3333    13.3332
1.8340    5.0
2.0
0.0       13.3332
0.0       0.0
2.0
0.0       13.3332
0.0       0.0
1.0
110.0     0.0       13.3332
3.0       1.0       0.0       13.3332   26.0430
-.9170    0.0       0.0       3.0       0.0       0.0
2.0
13.3332   26.0430
5.0       5.0
2.0
13.3332   26.0430
0.0       0.0
2.0
13.3332   26.0430
0.0       0.0
1.0
110.0     13.3332   26.0430
8.0       4.0       0.0       26.0430   62.4172
-.9170    0.0       0.0       3.0       0.0       0.0
5.0
26.0430   40.0815   42.9588   42.9982   62.4172
5.0       4.1667    3.9173    3.9402    2.1883
2.0
26.0430   62.4172
0.0       0.0
2.0
26.0430   62.4172
0.0       0.0
1.0
110.0     26.0430   62.4172
1.0       1.0       0.0       62.4172   63.9592
-.9170    0.0       0.0       3.0       0.0       0.0
2.0
62.4172   63.9592
2.1883    1.2254
2.0
62.4172   63.9592
0.0       0.0
2.0
62.4172   63.9592
0.0       0.3043
1.0
110.0     62.4172   63.9592
1.0       3.0
```

```
110.0     NACA 23024 *** ABBOTT & VON DOENHOFF *** CMA STALL MODEL
1.0       13.0      1.0       13.0      1.0       2.0
0.24      32.0      1.0       1.0       0.0       2.0       0.5960
          3.0000
-4.0      -.3000
-2.0      -.1100
0.0       0.0950
2.0       0.2800
4.0       0.4400
6.0       0.6250
8.0       0.8100
9.0       0.8900
10.0      0.9750
11.0      1.0500
12.0      1.1250
13.0      1.1800
14.0      1.1950
          3.0000
-4.0      0.0084
-2.0      0.0082
0.0       0.0084
2.0       0.0088
4.0       0.0096
6.0       0.0106
8.0       0.0121
9.0       0.0134
10.0      0.0153
11.0      0.0161
12.0      0.0163
13.0      0.0183
14.0      0.0250
          0.0000
-180.0    0.0
180.0     0.0
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

MOD-0 (AILERON) *** CMA STALL MODEL *** 05

REFERENCE BLADE SURFACE PANEL COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10	11
1	X	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337
	Y	.052045	.072963	.130112	.208178	.208178	.274327	.340475	.406623	.406623	.517421	.623961
	Z	.002728	.003824	.006819	.010910	.010910	.014377	.017844	.021310	.021310	.027117	.032700
2	X	-.005940	-.003998	.001308	.008557	.008557	.008557	.008557	.008557	.008557	.006628	.004773
	Y	.052052	.072971	.130125	.208198	.208198	.274346	.340494	.406643	.406643	.517439	.623977
	Z	.002592	.003657	.006566	.010540	.010540	.014007	.017473	.020940	.020940	.026778	.032391
3	X	.014334	.020964	.039080	.063827	.063827	.063827	.063827	.063827	.063827	.057242	.050911
	Y	.052069	.072992	.130157	.208245	.208245	.274393	.340541	.406690	.406690	.517482	.624016
	Z	.002264	.003253	.005955	.009646	.009646	.013113	.016580	.020046	.020046	.025959	.031645

I,J		12	13	14	15	16	17	18	19
1	X	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337	-.014337	-.009580
	Y	.722148	.808210	.878840	.931322	.963641	.974553	.974553	.998634
	Z	.037846	.042357	.046058	.048809	.050502	.051074	.051074	.052259
2	X	.002360	.000083	-.001785	-.003174	-.004029	-.004318	-.004318	-.003969
	Y	.722162	.808223	.878851	.931332	.963649	.974562	.974562	.998638
	Z	.037576	.042123	.045855	.048628	.050336	.050912	.050912	.052168
3	X	.042671	.034898	.028518	.023777	.020858	.019872	.019872	.009576
	Y	.722197	.808252	.878876	.931355	.963671	.974582	.974582	.998650
	Z	.036924	.041560	.045365	.048192	.049933	.050521	.050521	.051949

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

MOD-0 (AILERON) *** CMA STALL MODEL *** 05

REFERENCE BLADE COLLOCATION POINT COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10
1	X	-.011090	-.009834	-.007659	-.006402	-.006402	-.006402	-.006737	-.007392	-.008100	-.008945
	Y	.062506	.101541	.169151	.241259	.307408	.373556	.462029	.570697	.673060	.765184
	Z	.003223	.005249	.008757	.012515	.015982	.019449	.024091	.029796	.035172	.040014
2	X	.004098	.011229	.023581	.030712	.030712	.030712	.028814	.025092	.021075	.016279
	Y	.062519	.101559	.169177	.241291	.307439	.373587	.462059	.570724	.673085	.765205
	Z	.002978	.004908	.008251	.011915	.015382	.018848	.023516	.029271	.034701	.039606

I,J		11	12	13	14	15
1	X	-.009663	-.010228	-.010616	-.010815	-.009250
	Y	.843529	.905085	.947485	.969100	.986596
	Z	.044132	.047367	.049595	.050731	.051623
2	X	.012200	.008996	.006788	.005663	.003420
	Y	.843548	.905101	.947499	.969114	.986607
	Z	.043778	.047056	.049314	.050465	.051418

```
COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105
-------------------------------------------------------------------------------------------------------------------------

POINT  8: MOD-0 (AILERON) *** CMA STALL MODEL *** 05
ROTOR PITCH       (DEG)       .0000
WIND SPEED      (M/SEC)     10.0000
ROTOR AZIMUTH     (DEG)       .0000

WORKING STATE AND WAKE RATE PARAMETERS
--------------------------------------

WINDMILL BRAKE STATE

  REFERRED VELOCITIES,
    AXIAL ONSET    4.05452E+00
    INDUCED       -2.63803E-01

  REFERENCE WAKE PARAMETERS,
    CTRW           6.87899E-03
    AKZO          -8.29397E-02
    ALAMO          3.31759E-01
    AKRO          -4.00000E+00

  MODIFIED WAKE PARAMETERS,
    AKZW           2.37786E-01
    AKZI          -1.26566E-02
    AKZ            2.25129E-01
    AKR            1.47364E+00
    A              1.22000E+00
    AKB            2.71438E+00
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 8: MOD-0 (AILERON) *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 10.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

GEOMETRY

ELEMENT NUMBER		1	2	3	4	5	6	7	8
ELE CENTER STATION		6.250E-02	1.015E-01	1.691E-01	2.413E-01	3.074E-01	3.735E-01	4.620E-01	5.707E-01
CHORD	FT	2.046E+00	2.838E+00	4.209E+00	5.000E+00	5.000E+00	5.000E+00	4.789E+00	4.376E+00
ELEMENT SPAN	FT	1.340E+00	3.660E+00	5.000E+00	4.237E+00	4.237E+00	4.237E+00	7.096E+00	6.824E+00
LEADING EDGE SWEEP	DEG	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
AERODYNAMIC SWEEP	DEG	3.554E+00	2.603E+00	1.269E+00	2.179E+00	1.913E+00	1.679E+00	2.003E+00	1.531E+00

ANGLES OF ATTACK

GEOMETRIC PITCH	DEG	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
ZERO LIFT LINE	DEG	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01
EFFECTIVE ANGLE	DEG	6.586E+01	6.082E+01	5.101E+01	4.199E+01	3.516E+01	3.034E+01	2.557E+01	2.122E+01
WIND ANGLE	DEG	7.526E+01	6.688E+01	5.465E+01	4.470E+01	3.784E+01	3.259E+01	2.732E+01	2.268E+01
INFLOW ANGLE	DEG	7.447E+01	6.679E+01	5.461E+01	4.434E+01	3.682E+01	3.159E+01	2.643E+01	2.178E+01
INDUCED ANGLE	DEG	7.318E-01	4.129E+00	1.929E+00	3.002E-01	-7.850E-01	-7.830E-01	-7.275E-01	-7.892E-01

VELOCITIES

INDUCED VELOCITY	FT/SEC	5.129E-01	5.107E-01	6.062E-01	6.092E-01	7.966E-01	8.713E-01	8.550E-01	8.621E-01
RELATIVE VELOCITY	FT/SEC	3.417E+01	3.686E+01	4.136E+01	4.743E+01	5.402E+01	6.156E+01	7.217E+01	8.574E+01
REYNOLDS NUMBER		4.471E+05	6.688E+05	1.113E+06	1.516E+06	1.727E+06	1.968E+06	2.210E+06	2.399E+06
RELATIVE Q	PSF	1.388E+00	1.615E+00	2.034E+00	2.675E+00	3.469E+00	4.506E+00	6.192E+00	8.740E+00

AERODYNAMIC COEFFICIENTS

ELE LIFT COEF		4.557E-01	5.162E-01	5.864E-01	7.029E-01	8.576E-01	9.125E-01	9.476E-01	1.004E+00
ELE DRAG COEF		9.725E-01	8.893E-01	7.006E-01	5.854E-01	5.671E-01	4.402E-01	2.572E-01	1.349E-01
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00

FORCE LOADING

ELE X FORCE	LBS/FT	-1.786E+03	-2.324E+03	-2.531E+03	-3.968E+03	-4.259E+03	-9.490E+03	-2.323E+04	-3.871E+04
ELE IND X FORCE	LBS/FT	-4.811E+03	-8.896E+03	-1.674E+04	-2.688E+04	-3.647E+04	-4.405E+04	-5.117E+04	-5.831E+04
ELE PRO X FORCE	LBS/FT	3.025E+03	6.571E+03	1.421E+04	2.291E+04	3.221E+04	3.456E+04	2.794E+04	1.960E+04
ELE Z FORCE	LBS/FT	1.228E+04	1.911E+04	3.185E+04	4.982E+04	7.273E+04	9.277E+04	1.167E+05	1.535E+05
ELE IND Z FORCE	LBS/FT	1.411E+03	3.809E+03	1.188E+04	2.747E+04	4.864E+04	7.154E+04	1.028E+05	1.457E+05
ELE PRO Z FORCE	LBS/FT	1.087E+04	1.530E+04	1.997E+04	2.236E+04	2.408E+04	2.122E+04	1.387E+04	7.824E+03

TORQUE LOADING

ELE PRO TORQUE	FT-LBS/FT	1.209E+04	4.267E+04	1.537E+05	3.535E+05	6.333E+05	8.258E+05	8.257E+05	7.155E+05
ELE IND TORQUE	FT-LBS/FT	-1.923E+04	-5.777E+04	-1.811E+05	-4.147E+05	-7.170E+05	-1.052E+06	-1.512E+06	-2.128E+06
ELE TORQUE	FT-LBS/FT	-7.141E+03	-1.510E+04	-2.738E+04	-6.122E+04	-8.373E+04	-2.267E+05	-6.865E+05	-1.413E+06

PITCHING MOMENT LOADING (2D)

ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00

LIFT SLOPE CONVERGENCE

INPUT LIFT SLOPE	/RAD	4.992E-01	5.911E-01	7.538E-01	1.056E+00	1.490E+00	1.803E+00	2.194E+00	2.775E+00
CL/ALPHAE	/RAD	4.993E-01	5.912E-01	7.545E-01	1.051E+00	1.489E+00	1.806E+00	2.196E+00	2.774E+00

CIRCULATION PARAMETERS

CIRC SHAPE PARM		8.260E-01	8.243E-01	8.284E-01	8.383E-01	8.532E-01	8.623E-01	8.743E-01	8.926E-01
BOUND CIRCULATION	FT**2/SEC	1.591E+01	2.697E+01	5.098E+01	8.375E+01	1.158E+02	1.401E+02	1.635E+02	1.884E+02

COMPUTATIONAL METHODOLOGY ASSOCIATES *** ISWT VERSION 1.00 RELEASED MAY 1986 *** SERI II S 05105

POINT 8: MOD-0 (AILERON) *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 10.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

GEOMETRY

ELEMENT NUMBER		9	10	11	12	13	14	15
ELE CENTER STATION		6.731E-01	7.652E-01	8.475E-01	9.051E-01	9.479E-01	9.691E-01	9.866E-01
CHORD	FT	3.930E+00	3.398E+00	2.945E+00	2.590E+00	2.345E+00	2.220E+00	1.707E+00
ELEMENT SPAN	FT	6.289E+00	5.512E+00	4.524E+00	3.361E+00	2.070E+00	6.989E-01	1.542E+00
LEADING EDGE SWEEP	DEG	.000E+00	.000E+00	.000E+00	1.978E-02	.000E+00	.000E+00	1.116E+01
AERODYNAMIC SWEEP	DEG	1.496E+00	1.256E+00	1.029E+00	8.767E-01	7.841E-01	7.420E-01	2.924E+00

ANGLES OF ATTACK

GEOMETRIC PITCH	DEG	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
ZERO LIFT LINE	DEG	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01	-9.277E-01
EFFECTIVE ANGLE	DEG	1.814E+01	1.599E+01	1.448E+01	1.344E+01	1.230E+01	1.081E+01	1.075E+01
WIND ANGLE	DEG	1.950E+01	1.729E+01	1.575E+01	1.472E+01	1.408E+01	1.378E+01	1.354E+01
INFLOW ANGLE	DEG	1.851E+01	1.624E+01	1.463E+01	1.352E+01	1.230E+01	1.072E+01	1.061E+01
INDUCED ANGLE	DEG	-9.135E-01	-9.885E-01	-1.068E+00	-1.158E+00	-1.760E+00	-3.096E+00	-2.941E+00

VELOCITIES

INDUCED VELOCITY	FT/SEC	9.143E-01	9.372E-01	9.687E-01	9.928E-01	1.063E+00	1.004E+00	1.133E+00
RELATIVE VELOCITY	FT/SEC	9.820E+01	1.109E+02	1.213E+02	1.295E+02	1.349E+02	1.371E+02	1.397E+02
REYNOLDS NUMBER		2.485E+06	2.410E+06	2.284E+06	2.144E+06	2.022E+06	1.946E+06	1.524E+06
RELATIVE Q	PSF	1.163E+01	1.463E+01	1.749E+01	1.993E+01	2.164E+01	2.235E+01	2.319E+01

AERODYNAMIC COEFFICIENTS

ELE LIFT COEF		1.073E+00	1.144E+00	1.188E+00	1.153E+00	1.078E+00	9.652E-01	9.553E-01
ELE DRAG COEF		7.185E-02	3.859E-02	2.200E-02	1.731E-02	1.617E-02	1.508E-02	1.488E-02
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00

FORCE LOADING

ELE X FORCE	LBS/FT	-5.094E+04	-5.766E+04	-5.877E+04	-5.334E+04	-4.438E+04	-3.343E+04	-2.612E+04
ELE IND X FORCE	LBS/FT	-6.368E+04	-6.520E+04	-6.325E+04	-5.629E+04	-4.766E+04	-3.644E+04	-2.849E+04
ELE PRO X FORCE	LBS/FT	1.274E+04	7.534E+03	4.486E+03	3.554E+03	3.279E+03	3.008E+03	2.368E+03
ELE Z FORCE	LBS/FT	1.943E+05	2.258E+05	2.431E+05	2.372E+05	2.189E+05	1.928E+05	1.525E+05
ELE IND Z FORCE	LBS/FT	1.900E+05	2.236E+05	2.412E+05	2.363E+05	2.182E+05	1.922E+05	1.518E+05
ELE PRO Z FORCE	LBS/FT	4.259E+03	2.192E+03	1.170E+03	8.534E+02	7.142E+02	5.486E+02	4.432E+02

TORQUE LOADING

ELE PRO TORQUE	FT-LBS/FT	5.424E+05	5.488E+05	5.420E+05	5.057E+05	4.287E+05	3.864E+05	1.424E+05
ELE IND TORQUE	FT-LBS/FT	-2.741E+06	-3.191E+06	-3.413E+06	-3.292E+06	-2.888E+06	-2.259E+06	-1.572E+06
ELE TORQUE	FT-LBS/FT	-2.193E+06	-2.822E+06	-3.171E+06	-3.088E+06	-2.690E+06	-2.072E+06	-1.448E+06

PITCHING MOMENT LOADING (2D)

ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00

LIFT SLOPE CONVERGENCE

INPUT LIFT SLOPE	/RAD	3.470E+00	4.181E+00	4.739E+00	4.958E+00	5.059E+00	5.144E+00	5.133E+00
CL/ALPHAE	/RAD	3.447E+00	4.161E+00	4.751E+00	4.961E+00	5.059E+00	5.144E+00	5.133E+00

CIRCULATION PARAMETERS

CIRC SHAPE PARM		9.157E-01	9.371E-01	9.549E-01	9.626E-01	9.711E-01	9.858E-01	9.211E-01
BOUND CIRCULATION	FT**2/SEC	2.099E+02	2.171E+02	2.116E+02	1.931E+02	1.704E+02	1.429E+02	1.157E+02

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 8: MOD-0 (AILERON) *** CMA STALL MODEL *** 05

ROTOR PITCH	(DEG)	.0000
WIND SPEED	(M/SEC)	10.0000
ROTOR AZIMUTH	(DEG)	.0000

ROTOR PERFORMANCE

THRUST	LBS	4.0183E+03
CT ROTOR		6.9068E-03
CT WT		2.4431E-01
TORQUE	FT-LBS	-3.9070E+04
CQ ROTOR		-1.0500E-03
CP WT		-1.5619E-01
POWER (ROTOR)	KW	-1.1427E+02
POWER (PROFILE)	KW	3.9184E+01
POWER (INFLOW)	KW	-1.5346E+02

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

RECORD IMAGE OF INPUT FILE

```
ESI-54 *** CMA STALL MODEL *** 05
3.0       0.0       2.0       0.0       2.0       2.0       0.0
1.0
0.0       0.0       0.0
0.0       0.0       0.0       0.0
0.0       77.0      1.0       1.0
5.0       1.0       18.0      1.0
0.0       0.0       1.0       600.0
2.0       2.0       2.0       27.0
2.0       12.0      0.0       180.0     30.0      1.0
0.0       0.0       1.0       13.0      14.0      17.0
4.0       1.0       0.25      2.70      8.10
0.0       0.0       0.0       5.0       0.0       0.0
2.0
0.0       8.10
1.15      2.05
2.0
0.0       27.0
-7.5      2.5
2.0
0.0       27.0
0.0       0.0
1.0
120.0     0.0       27.0
11.0      4.0       0.25      8.10      27.0
0.0       0.0       0.0       5.0       0.0       0.0
2.0
8.10      27.0
2.05      1.0
2.0
0.0       27.0
-7.5      2.5
2.0
0.0       27.0
0.0       0.0
1.0
120.0     0.0       27.0
1.0       3.0
```

```
120.0     NACA 23012 *** VDT1167 TR586 *** CMA STALL MODEL
8.0       14.0       8.0        14.0       1.0        2.0
0.12      32.0       1.0        1.0        0.0        2.0        0.5960
          0.0424     0.0842     0.1700     0.3350     0.6650     1.2860
          2.3000     3.0900
-2.0      -0.030     -0.030     -0.080     -0.080     -0.080     -0.080
          -0.080     -0.080
0.0       0.1600     0.1400     0.1300     0.1300     0.1300     0.1300
          0.1300     0.1300
2.0       0.4000     0.4000     0.3800     0.3300     0.3300     0.3300
          0.3300     0.3300
4.0       0.5800     0.5800     0.5400     0.5400     0.5400     0.5400
          0.5400     0.5400
6.0       0.7400     0.7400     0.7200     0.7200     0.7200     0.7200
          0.7200     0.7200
8.0       0.8600     0.9000     0.9000     0.9000     0.9300     0.9300
          0.9300     0.9300
9.0       0.9250     0.9800     0.9600     0.9800     1.0500     1.0600
          1.0600     1.0600
10.0      0.9000     1.0200     1.0600     1.0800     1.1100     1.1500
          1.1500     1.1500
11.0      0.8600     1.0800     1.0900     1.1500     1.1900     1.2300
          1.2400     1.2400
12.0      0.8400     0.9900     1.1000     1.1800     1.2500     1.3000
          1.3100     1.3200
13.0      0.8200     0.9400     1.1000     1.2000     1.3000     1.3800
          1.3900     1.4100
14.0      0.8000     0.9000     1.0300     1.1900     1.3000     1.4000
          1.4700     1.4900
15.0      0.7900     0.8800     0.9750     1.1200     1.2700     1.4200
          1.5400     1.5700
16.0      0.7800     0.8600     0.9400     1.0800     1.2200     1.2700
          1.5600     1.6000
          0.0424     0.0842     0.1700     0.3350     0.6650     1.2860
          2.3000     3.0900
-2.0      0.0265     0.0258     0.0125     0.0120     0.0120     0.0100
          0.0100     0.0090
0.0       0.0220     0.0215     0.0125     0.0115     0.0115     0.0100
          0.0100     0.0085
2.0       0.0225     0.0210     0.0125     0.0115     0.0115     0.0100
          0.0100     0.0080
4.0       0.0230     0.0215     0.0135     0.0120     0.0200     0.0100
          0.0100     0.0085
6.0       0.0270     0.0250     0.0170     0.0140     0.0140     0.0120
          0.0120     0.0105
8.0       0.0335     0.0313     0.0200     0.0170     0.0165     0.0135
          0.0135     0.0120
9.0       0.0424     0.0350     0.0225     0.0190     0.0180     0.0150
          0.0155     0.0140
10.0      0.0539     0.0400     0.0280     0.0230     0.0195     0.0175
          0.0170     0.0155
11.0      0.0666     0.0500     0.0400     0.0290     0.0240     0.0210
          0.0185     0.0175
12.0      0.0804     0.0643     0.0530     0.0350     0.0290     0.0240
          0.0215     0.0215
13.0      0.0954     0.0797     0.0688     0.0480     0.0400     0.0385
          0.0245     0.0235
14.0      0.1114     0.0963     0.0858     0.0657     0.0579     0.0335
          0.0295     0.0275
15.0      0.1286     0.1140     0.1039     0.0845     0.0771     0.0500
          0.0400     0.0333
16.0      0.1468     0.1326     0.1231     0.1046     0.0975     0.0715
          0.0619     0.0500
          0.0
-180.0    0.0
180.0     0.0
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

ESI-54 *** CMA STALL MODEL *** 05

REFERENCE BLADE SURFACE PANEL COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10	11
1	X	-.013369	-.014764	-.016159	-.017555	-.018951	-.018951	-.017584	-.016238	-.014942	-.013725	-.012611
	Y	.099727	.149536	.199344	.249149	.298953	.298953	.398160	.495353	.588550	.675856	.755491
	Z	.007489	.011848	.016232	.020639	.025071	.025071	.034136	.042971	.051399	.059255	.066389
2	X	.002294	.002533	.002772	.003012	.003251	.003251	.003017	.002786	.002564	.002355	.002164
	Y	.099601	.149411	.199221	.249031	.298842	.298842	.398089	.495315	.588541	.675869	.755521
	Z	.008926	.013284	.017637	.021986	.026331	.026331	.034948	.043397	.051507	.059109	.066049
3	X	.040108	.044291	.048477	.052664	.056852	.056852	.052752	.048715	.044827	.041174	.037834
	Y	.099298	.149108	.198924	.248747	.298576	.298576	.397918	.495225	.588518	.675900	.755593
	Z	.012395	.016749	.021030	.025238	.029374	.029374	.036908	.044427	.051767	.058756	.065229

I,J		12	13	14	15	16	17
1	X	-.011627	-.010791	-.010123	-.009636	-.009339	-.009240
	Y	.825834	.885452	.933130	.967897	.989045	.996142
	Z	.072664	.077964	.082190	.085264	.087131	.087757
2	X	.001995	.001851	.001737	.001653	.001602	.001585
	Y	.825876	.885502	.933186	.967956	.989106	.996204
	Z	.072184	.077386	.081549	.084585	.086432	.087053
3	X	.034880	.032374	.030369	.028907	.028017	.027719
	Y	.825978	.885624	.933321	.968100	.989253	.996353
	Z	.071024	.075991	.080001	.082946	.084746	.085352

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

ESI-54 *** CMA STALL MODEL *** 05

REFERENCE BLADE COLLOCATION POINT COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10
1	X	-.008355	-.009183	-.010012	-.010841	-.010850	-.010045	-.009260	-.008514	-.007821	-.007198
	Y	.124588	.174397	.224205	.274012	.348525	.446737	.541943	.632203	.715681	.790675
	Z	.010165	.014530	.018911	.023305	.029966	.038772	.047281	.055323	.062740	.069386
2	X	.018362	.020182	.022004	.023826	.023847	.022076	.020352	.018711	.017189	.015820
	Y	.124384	.174195	.224009	.273826	.348379	.446651	.541906	.632207	.715716	.790734
	Z	.012499	.016839	.021147	.025424	.031635	.039765	.047704	.055283	.062339	.068716

I,J		11	12	13	14	15
1	X	-.006658	-.006211	-.005868	-.005635	-.005517
	Y	.855659	.909309	.950534	.978492	.992615
	Z	.075132	.079866	.083498	.085959	.087201
2	X	.014632	.013650	.012896	.012384	.012126
	Y	.855734	.909396	.950627	.978589	.992714
	Z	.074270	.078876	.082427	.084842	.086064

```
COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED JANUARY 1986 *** (PRELIMINARY RELEASE)
-------------------------------------------------------------------------------------------------------------

POINT  8: ESI-54 *** CMA STALL MODEL *** 05
ROTOR PITCH       (DEG)       .0000
WIND SPEED      (M/SEC)     12.0000
ROTOR AZIMUTH     (DEG)       .0000

WORKING STATE AND WAKE RATE PARAMETERS
--------------------------------------

WINDMILL BRAKE STATE

  REFERRED VELOCITIES,
    AXIAL ONSET    3.07276E+00
    INDUCED       -3.69991E-01

  REFERENCE WAKE PARAMETERS,
    CTRW           6.92691E-03
    AKZO          -8.32280E-02
    ALAMO          3.32912E-01
    AKRO          -4.00000E+00

  MODIFIED WAKE PARAMETERS,
    AKZW           1.80835E-01
    AKZI          -1.75745E-02
    AKZ            1.63261E-01
    AKR            2.03914E+00
    A              1.22000E+00
    AKB            1.96161E+00
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 8: ESI-54 *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 12.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

GEOMETRY

ELEMENT NUMBER		1	2	3	4	5	6	7	8
ELE CENTER STATION		1.245E-01	1.743E-01	2.241E-01	2.740E-01	3.485E-01	4.467E-01	5.419E-01	6.322E-01
CHORD	FT	1.525E+00	1.675E+00	1.825E+00	1.975E+00	1.975E+00	1.827E+00	1.684E+00	1.548E+00
ELEMENT SPAN	FT	1.350E+00	1.350E+00	1.350E+00	1.350E+00	2.690E+00	2.635E+00	2.527E+00	2.367E+00
LEADING EDGE SWEEP	DEG	1.591E+00	1.591E+00	1.591E+00	1.591E+00	7.957E-01	7.957E-01	7.955E-01	7.960E-01
AERODYNAMIC SWEEP	DEG	3.000E-01	6.256E-02	3.375E-01	5.439E-01	4.601E-01	2.121E-01	3.956E-02	7.913E-02

ANGLES OF ATTACK

GEOMETRIC PITCH	DEG	-6.250E+00	-5.750E+00	-5.250E+00	-4.750E+00	-4.002E+00	-3.016E+00	-2.060E+00	-1.154E+00
ZERO LIFT LINE	DEG	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00
EFFECTIVE ANGLE	DEG	4.678E+01	3.889E+01	3.247E+01	2.797E+01	2.254E+01	1.695E+01	1.473E+01	1.366E+01
WIND ANGLE	DEG	5.556E+01	4.618E+01	3.904E+01	3.358E+01	2.756E+01	2.213E+01	1.852E+01	1.601E+01
INFLOW ANGLE	DEG	5.564E+01	4.633E+01	3.881E+01	3.343E+01	2.689E+01	1.999E+01	1.668E+01	1.462E+01
INDUCED ANGLE	DEG	1.421E+00	1.373E+00	4.997E-01	4.740E-01	-3.677E-01	-2.055E+00	-1.759E+00	-1.320E+00

VELOCITIES

INDUCED VELOCITY	FT/SEC	3.796E-01	6.908E-01	9.374E-01	1.088E+00	1.229E+00	1.723E+00	1.667E+00	1.497E+00
RELATIVE VELOCITY	FT/SEC	4.867E+01	5.593E+01	6.380E+01	7.275E+01	8.643E+01	1.053E+02	1.249E+02	1.438E+02
REYNOLDS NUMBER		4.746E+05	5.990E+05	7.445E+05	9.187E+05	1.092E+06	1.231E+06	1.345E+06	1.423E+06
RELATIVE Q	PSF	2.817E+00	3.719E+00	4.840E+00	6.293E+00	8.881E+00	1.319E+01	1.855E+01	2.458E+01

AERODYNAMIC COEFFICIENTS

ELE LIFT COEF		5.959E-01	7.866E-01	9.061E-01	9.594E-01	1.065E+00	1.307E+00	1.392E+00	1.335E+00
ELE DRAG COEF		6.072E-01	5.777E-01	5.271E-01	3.468E-01	1.814E-01	6.757E-02	3.551E-02	2.908E-02
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00

FORCE LOADING

ELE X FORCE	LBS/FT	-2.190E+02	-7.721E+02	-1.012E+03	-2.167E+03	-4.090E+03	-6.732E+03	-8.325E+03	-8.568E+03
ELE IND X FORCE	LBS/FT	-1.292E+03	-2.584E+03	-3.657E+03	-4.789E+03	-6.159E+03	-7.848E+03	-9.100E+03	-9.348E+03
ELE PRO X FORCE	LBS/FT	1.073E+03	1.812E+03	2.645E+03	2.622E+03	2.070E+03	1.116E+03	7.746E+02	7.806E+02
ELE Z FORCE	LBS/FT	2.634E+03	4.348E+03	6.648E+03	8.951E+03	1.315E+04	2.189E+04	3.048E+04	3.590E+04
ELE IND Z FORCE	LBS/FT	1.071E+03	2.457E+03	4.529E+03	7.226E+03	1.210E+04	2.149E+04	3.025E+04	3.570E+04
ELE PRO Z FORCE	LBS/FT	1.564E+03	1.890E+03	2.119E+03	1.725E+03	1.046E+03	4.043E+02	2.312E+02	2.029E+02

TORQUE LOADING

ELE PRO TORQUE	FT-LBS/FT	3.607E+03	8.527E+03	1.601E+04	1.939E+04	1.947E+04	1.345E+04	1.133E+04	1.332E+04
ELE IND TORQUE	FT-LBS/FT	-4.344E+03	-1.216E+04	-2.213E+04	-3.543E+04	-5.795E+04	-9.465E+04	-1.332E+05	-1.596E+05
ELE TORQUE	FT-LBS/FT	-7.365E+02	-3.634E+03	-6.125E+03	-1.603E+04	-3.848E+04	-8.120E+04	-1.218E+05	-1.462E+05

PITCHING MOMENT LOADING (2D)

ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00

LIFT SLOPE CONVERGENCE

INPUT LIFT SLOPE	/RAD	8.174E-01	1.254E+00	1.682E+00	2.027E+00	2.794E+00	4.648E+00	5.460E+00	5.653E+00
CL/ALPHAE	/RAD	8.177E-01	1.253E+00	1.688E+00	2.046E+00	2.778E+00	4.483E+00	5.475E+00	5.654E+00

CIRCULATION PARAMETERS

CIRC SHAPE PARM		8.304E-01	8.400E-01	8.524E-01	8.594E-01	8.844E-01	9.478E-01	9.721E-01	9.771E-01
BOUND CIRCULATION	FT**2/SEC	2.210E+01	3.687E+01	5.256E+01	6.829E+01	9.143E+01	1.304E+02	1.460E+02	1.486E+02

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 8: ESI-54 *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 12.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

GEOMETRY

ELEMENT NUMBER		9	10	11	12	13	14	15
ELE CENTER STATION		7.157E-01	7.907E-01	8.557E-01	9.093E-01	9.506E-01	9.785E-01	9.926E-01
CHORD	FT	1.422E+00	1.309E+00	1.212E+00	1.131E+00	1.069E+00	1.027E+00	1.005E+00
ELEMENT SPAN	FT	2.159E+00	1.907E+00	1.616E+00	1.292E+00	9.424E-01	5.732E-01	1.924E-01
LEADING EDGE SWEEP	DEG	7.955E-01	7.957E-01	7.955E-01	7.955E-01	7.957E-01	7.957E-01	7.957E-01
AERODYNAMIC SWEEP	DEG	1.631E-01	2.247E-01	2.705E-01	3.033E-01	3.275E-01	3.443E-01	3.561E-01
ANGLES OF ATTACK								
GEOMETRIC PITCH	DEG	-3.157E-01	4.371E-01	1.090E+00	1.628E+00	2.042E+00	2.323E+00	2.464E+00
ZERO LIFT LINE	DEG	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00
EFFECTIVE ANGLE	DEG	1.297E+01	1.247E+01	1.206E+01	1.153E+01	1.056E+01	8.926E+00	5.676E+00
WIND ANGLE	DEG	1.422E+01	1.291E+01	1.195E+01	1.126E+01	1.078E+01	1.048E+01	1.033E+01
INFLOW ANGLE	DEG	1.305E+01	1.176E+01	1.067E+01	9.562E+00	8.111E+00	6.096E+00	2.504E+00
INDUCED ANGLE	DEG	-1.113E+00	-1.103E+00	-1.252E+00	-1.679E+00	-2.682E+00	-4.453E+00	-7.962E+00
VELOCITIES								
INDUCED VELOCITY	FT/SEC	1.384E+00	1.341E+00	1.346E+00	1.360E+00	1.198E+00	3.384E-01	-1.497E+00
RELATIVE VELOCITY	FT/SEC	1.614E+02	1.772E+02	1.909E+02	2.022E+02	2.103E+02	2.146E+02	2.148E+02
REYNOLDS NUMBER		1.468E+06	1.484E+06	1.479E+06	1.462E+06	1.437E+06	1.409E+06	1.381E+06
RELATIVE Q	PSF	3.096E+01	3.734E+01	4.335E+01	4.859E+01	5.257E+01	5.476E+01	5.488E+01
AERODYNAMIC COEFFICIENTS								
ELE LIFT COEF		1.283E+00	1.248E+00	1.217E+00	1.174E+00	1.089E+00	8.972E-01	5.794E-01
ELE DRAG COEF		2.275E-02	2.122E-02	1.997E-02	1.834E-02	1.583E-02	1.327E-02	1.044E-02
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
FORCE LOADING								
ELE X FORCE	LBS/FT	-8.587E+03	-8.329E+03	-7.877E+03	-7.088E+03	-5.650E+03	-3.364E+03	-5.985E+02
ELE IND X FORCE	LBS/FT	-9.299E+03	-9.070E+03	-8.628E+03	-7.812E+03	-6.291E+03	-3.905E+03	-1.018E+03
ELE PRO X FORCE	LBS/FT	7.116E+02	7.405E+02	7.516E+02	7.245E+02	6.417E+02	5.406E+02	4.194E+02
ELE Z FORCE	LBS/FT	4.014E+04	4.355E+04	4.578E+04	4.632E+04	4.407E+04	3.648E+04	2.321E+04
ELE IND Z FORCE	LBS/FT	3.998E+04	4.340E+04	4.564E+04	4.620E+04	4.398E+04	3.642E+04	2.319E+04
ELE PRO Z FORCE	LBS/FT	1.643E+02	1.536E+02	1.410E+02	1.216E+02	9.111E+01	5.751E+01	1.827E+01
TORQUE LOADING								
ELE PRO TORQUE	FT-LBS/FT	1.375E+04	1.581E+04	1.736E+04	1.779E+04	1.647E+04	1.428E+04	1.124E+04
ELE IND TORQUE	FT-LBS/FT	-1.797E+05	-1.936E+05	-1.993E+05	-1.918E+05	-1.615E+05	-1.032E+05	-2.728E+04
ELE TORQUE	FT-LBS/FT	-1.659E+05	-1.778E+05	-1.820E+05	-1.740E+05	-1.450E+05	-8.888E+04	-1.604E+04
PITCHING MOMENT LOADING (2D)								
ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
LIFT SLOPE CONVERGENCE								
INPUT LIFT SLOPE	/RAD	5.716E+00	5.779E+00	5.826E+00	5.872E+00	5.941E+00	5.782E+00	5.858E+00
CL/ALPHAE	/RAD	5.716E+00	5.780E+00	5.826E+00	5.872E+00	5.942E+00	5.782E+00	5.858E+00
CIRCULATION PARAMETERS								
CIRC SHAPE PARM		9.793E-01	9.823E-01	9.854E-01	9.907E-01	1.005E+00	1.039E+00	1.161E+00
BOUND CIRCULATION	FT**2/SEC	1.472E+02	1.448E+02	1.408E+02	1.342E+02	1.223E+02	9.883E+01	6.257E+01

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 8: ESI-54 *** CMA STALL MODEL *** 05

ROTOR PITCH	(DEG)	.0000
WIND SPEED	(M/SEC)	12.0000
ROTOR AZIMUTH	(DEG)	.0000

ROTOR PERFORMANCE

THRUST	LBS	1.7996E+03
CT ROTOR		6.9716E-03
CT WT		4.2638E-01
TORQUE	FT-LBS	-6.6231E+03
CQ ROTOR		-9.5026E-04
CP WT		-3.2138E-01
POWER (ROTOR)	KW	-7.2407E+01
POWER (PROFILE)	KW	1.0465E+01
POWER (INFLOW)	KW	-8.2873E+01

RECORD IMAGE OF INPUT FILE

```
CARTER 25 *** CMA STALL MODEL *** 05
3.0       0.0       2.0       0.0       2.0       2.0       0.0
1.0
0.0       0.0       0.0
0.0       0.0       0.0       0.0
0.0       124.0     1.0       1.0
5.0       1.0       18.0      1.0
0.0       1.0       1.0       300.0
2.0       2.0       2.0       16.0
2.0       12.0      0.0       180.0     30.0      1.0
0.0       0.0       1.0       14.0      15.0      17.0
9.0       3.0       0.25      1.280     10.40
0.0       0.0       0.0       6.0       0.0       0.0
8.0
0.0       0.80      2.40      4.00      5.60      7.20      8.80
10.40
3.6480    3.6480    3.0080    2.400     1.9040    1.4400    1.1520
1.0880
8.0
0.0       0.80      2.40      4.00      5.60      7.20      8.80
10.40
-41.00    - 34.00   -20.00    -11.00    -5.00     -1.200    0.60
1.00
2.0
0.0       16.0
0.0       0.0
1.0
120.0     0.0       16.0
6.0       4.0       0.25      10.40     16.0
0.0       0.0       0.0       6.0       0.0       0.0
2.0
10.40     16.0
1.0880    1.0880
2.0
10.40     16.0
1.00      1.00
2.0
0.0       16.0
0.0       0.0
1.0
120.0     0.0       16.0
1.0       3.0
```

```
120.0     NACA 23012 *** VDT1167 TR586 *** CMA STALL MODEL
8.0       14.0       8.0        14.0       1.0        2.0
0.12      32.0       1.0        1.0        0.0        2.0        0.5960
          0.0424     0.0842     0.1700     0.3350     0.6650     1.2860
          2.3000     3.0900
-2.0      -0.030     -0.030     -0.080     -0.080     -0.080     -0.080
          -0.080     -0.080
0.0       0.1600     0.1400     0.1300     0.1300     0.1300     0.1300
          0.1300     0.1300
2.0       0.4000     0.4000     0.3800     0.3300     0.3300     0.3300
          0.3300     0.3300
4.0       0.5800     0.5800     0.5400     0.5400     0.5400     0.5400
          0.5400     0.5400
6.0       0.7400     0.7400     0.7200     0.7200     0.7200     0.7200
          0.7200     0.7200
8.0       0.8600     0.9000     0.9000     0.9000     0.9300     0.9300
          0.9300     0.9300
9.0       0.9250     0.9800     0.9600     0.9800     1.0500     1.0600
          1.0600     1.0600
10.0      0.9000     1.0200     1.0600     1.0800     1.1100     1.1500
          1.1500     1.1500
11.0      0.8600     1.0800     1.0900     1.1500     1.1900     1.2300
          1.2400     1.2400
12.0      0.8400     0.9900     1.1000     1.1800     1.2500     1.3000
          1.3100     1.3200
13.0      0.8200     0.9400     1.1000     1.2000     1.3000     1.3800
          1.3900     1.4100
14.0      0.8000     0.9000     1.0300     1.1900     1.3000     1.4000
          1.4700     1.4900
15.0      0.7900     0.8800     0.9750     1.1200     1.2700     1.4200
          1.5400     1.5700
16.0      0.7800     0.8600     0.9400     1.0800     1.2200     1.2700
          1.5600     1.6000
          0.0424     0.0842     0.1700     0.3350     0.6650     1.2860
          2.3000     3.0900
-2.0      0.0265     0.0258     0.0125     0.0120     0.0120     0.0100
          0.0100     0.0090
0.0       0.0220     0.0215     0.0125     0.0115     0.0115     0.0100
          0.0100     0.0085
2.0       0.0225     0.0210     0.0125     0.0115     0.0115     0.0100
          0.0100     0.0080
4.0       0.0230     0.0215     0.0135     0.0120     0.0200     0.0100
          0.0100     0.0085
6.0       0.0270     0.0250     0.0170     0.0140     0.0140     0.0120
          0.0120     0.0105
8.0       0.0335     0.0313     0.0200     0.0170     0.0165     0.0135
          0.0135     0.0120
9.0       0.0424     0.0350     0.0225     0.0190     0.0180     0.0150
          0.0155     0.0140
10.0      0.0539     0.0400     0.0280     0.0230     0.0195     0.0175
          0.0170     0.0155
11.0      0.0666     0.0500     0.0400     0.0290     0.0240     0.0210
          0.0185     0.0175
12.0      0.0804     0.0643     0.0530     0.0350     0.0290     0.0240
          0.0215     0.0215
13.0      0.0954     0.0797     0.0688     0.0480     0.0400     0.0385
          0.0245     0.0235
14.0      0.1114     0.0963     0.0858     0.0657     0.0579     0.0335
          0.0295     0.0275
15.0      0.1286     0.1140     0.1039     0.0845     0.0771     0.0500
          0.0400     0.0333
16.0      0.1468     0.1326     0.1231     0.1046     0.0975     0.0715
          0.0619     0.0500
          0.0
-180.0    0.0
180.0     0.0
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

CARTER 25 *** CMA STALL MODEL *** 05

REFERENCE BLADE SURFACE PANEL COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10	11
1	X	-.047428	-.047195	-.046281	-.044077	-.039921	-.034577	-.028618	-.022273	-.017980	-.016987	-.016987
	Y	.082260	.090725	.115877	.157023	.213154	.282549	.363167	.452551	.547942	.646370	.646370
	Z	-.017314	-.015009	-.008296	.001942	.013092	.024969	.036566	.047615	.058173	.068604	.068604
2	X	.008137	.008097	.007941	.007562	.006849	.005932	.004910	.003821	.003085	.002915	.002915
	Y	.079099	.087736	.113383	.155249	.212020	.281973	.362972	.452557	.548012	.646451	.646451
	Z	.012768	.013433	.015430	.018816	.023882	.030448	.038425	.047557	.057499	.067830	.067830
3	X	.142285	.141584	.138842	.132231	.119764	.103730	.085854	.066818	.053941	.050961	.050961
	Y	.071466	.080519	.107363	.150968	.209282	.280583	.362500	.452571	.548183	.646647	.646647
	Z	.085391	.082098	.072710	.059553	.049931	.043676	.042914	.047416	.055872	.065963	.065963

I,J		12	13	14	15	16	17
1	X	-.016987	-.016987	-.016987	-.016987	-.016987	-.016987
	Y	.736460	.820411	.892501	.947818	.982592	.994453
	Z	.078073	.086896	.094473	.100287	.103942	.105189
2	X	.002915	.002915	.002915	.002915	.002915	.002915
	Y	.736542	.820492	.892583	.947900	.982673	.994534
	Z	.077299	.086123	.093700	.099514	.103169	.104415
3	X	.050961	.050961	.050961	.050961	.050961	.050961
	Y	.736738	.820689	.892779	.948096	.982869	.994730
	Z	.075431	.084255	.091832	.097646	.101301	.102548

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

CARTER 25 *** CMA STALL MODEL *** 05

REFERENCE BLADE COLLOCATION POINT COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10
1	X	-.028103	-.027784	-.026907	-.024990	-.022026	-.018772	-.015080	-.011944	-.010384	-.010089
	Y	.085427	.102350	.135698	.184578	.247523	.322711	.407819	.500257	.597182	.691443
	Z	-.006018	-.002597	.003980	.012373	.022151	.032166	.042472	.052787	.063138	.073070
2	X	.061765	.061062	.059134	.054921	.048407	.041256	.033143	.026251	.022823	.022174
	Y	.080441	.097910	.132280	.182242	.246239	.322129	.407689	.500326	.597306	.691575
	Z	.041415	.039641	.036502	.034598	.034374	.037708	.043711	.052129	.061962	.071816

I,J		11	12	13	14	15
1	X	-.010089	-.010089	-.010089	-.010089	-.010089
	Y	.778464	.856484	.920188	.965233	.988550
	Z	.082216	.090417	.097112	.101847	.104297
2	X	.022174	.022174	.022174	.022174	.022174
	Y	.778596	.856616	.920320	.965365	.988682
	Z	.080962	.089163	.095858	.100593	.103043

```
COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105
-------------------------------------------------------------------------------------------------------------------------

POINT  8: CARTER 25 *** CMA STALL MODEL *** 05
ROTOR PITCH       (DEG)       .0000
WIND SPEED      (M/SEC)     12.0000
ROTOR AZIMUTH     (DEG)       .0000

WORKING STATE AND WAKE RATE PARAMETERS
--------------------------------------

WINDMILL BRAKE STATE

  REFERRED VELOCITIES,
    AXIAL ONSET    2.73829E+00
    INDUCED       -4.33967E-01

  REFERENCE WAKE PARAMETERS,
    CTRW           9.57775E-03
    AKZO          -9.78660E-02
    ALAMO          3.91464E-01
    AKRO          -4.00000E+00

  MODIFIED WAKE PARAMETERS,
    AKZW           1.89494E-01
    AKZI          -2.39673E-02
    AKZ            1.65527E-01
    AKR            2.36496E+00
    A              1.22000E+00
    AKB            1.69136E+00
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 8: CARTER 25 *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 12.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

		1	2	3	4	5	6	7	8
GEOMETRY									
ELEMENT NUMBER									
ELE CENTER STATION		8.387E-02	1.010E-01	1.346E-01	1.838E-01	2.471E-01	3.225E-01	4.078E-01	5.003E-01
CHORD	FT	3.428E+00	3.318E+00	3.102E+00	2.796E+00	2.409E+00	2.031E+00	1.625E+00	1.287E+00
ELEMENT SPAN	FT	1.386E-01	4.114E-01	6.718E-01	9.118E-01	1.124E+00	1.302E+00	1.441E+00	1.536E+00
LEADING EDGE SWEEP	DEG	5.662E+00	5.663E+00	5.608E+00	5.409E+00	4.914E+00	4.368E+00	4.056E+00	2.555E+00
AERODYNAMIC SWEEP	DEG	9.369E+00	8.591E+00	7.220E+00	5.088E+00	3.358E+00	1.970E+00	8.185E-01	1.065E-01
ANGLES OF ATTACK									
GEOMETRIC PITCH	DEG	-2.919E+01	-2.679E+01	-2.205E+01	-1.686E+01	-1.114E+01	-6.542E+00	-2.719E+00	-2.454E-01
ZERO LIFT LINE	DEG	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00
EFFECTIVE ANGLE	DEG	6.913E+00	1.090E+01	1.465E+01	2.449E+01	2.342E+01	2.006E+01	2.006E+01	1.976E+01
WIND ANGLE	DEG	6.726E+01	6.302E+01	5.553E+01	4.660E+01	3.800E+01	3.078E+01	2.513E+01	2.086E+01
INFLOW ANGLE	DEG	3.554E+01	3.747E+01	3.670E+01	4.182E+01	3.495E+01	2.679E+01	2.297E+01	2.017E+01
INDUCED ANGLE	DEG	-3.224E+01	-2.749E+01	-2.091E+01	-4.342E+00	-2.419E+00	-3.858E+00	-1.935E+00	-4.413E-01
VELOCITIES									
INDUCED VELOCITY	FT/SEC	1.305E+01	1.078E+01	8.059E+00	3.384E+00	2.765E+00	2.707E+00	2.112E+00	1.671E+00
RELATIVE VELOCITY	FT/SEC	3.725E+01	3.982E+01	4.481E+01	5.566E+01	6.595E+01	7.807E+01	9.430E+01	1.125E+02
REYNOLDS NUMBER		8.166E+05	8.449E+05	8.886E+05	9.951E+05	1.016E+06	1.014E+06	9.802E+05	9.261E+05
RELATIVE Q	PSF	1.650E+00	1.885E+00	2.387E+00	3.684E+00	5.171E+00	7.248E+00	1.057E+01	1.505E+01
AERODYNAMIC COEFFICIENTS									
ELE LIFT COEF		6.724E-01	1.074E+00	1.311E+00	1.010E+00	1.038E+00	1.131E+00	1.131E+00	1.138E+00
ELE DRAG COEF		1.385E-02	1.791E-02	4.257E-02	2.326E-01	2.050E-01	1.299E-01	1.314E-01	1.278E-01
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
FORCE LOADING									
ELE X FORCE	LBS/FT	-4.390E+02	-1.024E+03	-1.420E+03	-1.319E+03	-1.360E+03	-1.485E+03	-1.409E+03	-1.351E+03
ELE IND X FORCE	LBS/FT	-4.553E+02	-1.047E+03	-1.485E+03	-1.776E+03	-1.896E+03	-1.922E+03	-1.941E+03	-1.946E+03
ELE PRO X FORCE	LBS/FT	1.632E+01	2.277E+01	6.470E+01	4.570E+02	5.358E+02	4.371E+02	5.323E+02	5.950E+02
ELE Z FORCE	LBS/FT	8.581E+02	1.375E+03	2.029E+03	2.381E+03	3.070E+03	4.005E+03	4.779E+03	5.486E+03
ELE IND Z FORCE	LBS/FT	8.465E+02	1.358E+03	1.981E+03	1.974E+03	2.698E+03	3.786E+03	4.555E+03	5.269E+03
ELE PRO Z FORCE	LBS/FT	1.159E+01	1.736E+01	4.796E+01	4.067E+02	3.724E+02	2.195E+02	2.243E+02	2.173E+02
TORQUE LOADING									
ELE PRO TORQUE	FT-LBS/FT	2.189E+01	3.678E+01	1.394E+02	1.344E+03	2.119E+03	2.256E+03	3.473E+03	4.762E+03
ELE IND TORQUE	FT-LBS/FT	-6.109E+02	-1.691E+03	-3.198E+03	-5.225E+03	-7.495E+03	-9.920E+03	-1.267E+04	-1.558E+04
ELE TORQUE	FT-LBS/FT	-5.890E+02	-1.654E+03	-3.059E+03	-3.881E+03	-5.377E+03	-7.664E+03	-9.192E+03	-1.081E+04
PITCHING MOMENT LOADING (2D)									
ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
LIFT SLOPE CONVERGENCE									
INPUT LIFT SLOPE	/RAD	5.568E+00	5.627E+00	5.038E+00	2.630E+00	2.745E+00	3.445E+00	3.411E+00	3.466E+00
CL/ALPHAE	/RAD	5.587E+00	5.678E+00	5.182E+00	2.437E+00	2.610E+00	3.299E+00	3.296E+00	3.366E+00
CIRCULATION PARAMETERS									
CIRC SHAPE PARM		1.196E+00	1.147E+00	1.049E+00	8.703E-01	8.722E-01	9.044E-01	9.035E-01	9.029E-01
BOUND CIRCULATION	FT**2/SEC	4.223E+01	6.955E+01	8.786E+01	8.450E+01	8.656E+01	9.365E+01	8.966E+01	8.485E+01

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 8: CARTER 25 *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 12.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

GEOMETRY

ELEMENT NUMBER		9	10	11	12	13	14	15
ELE CENTER STATION		5.972E-01	6.915E-01	7.785E-01	8.565E-01	9.202E-01	9.653E-01	9.886E-01
CHORD	FT	1.120E+00	1.088E+00	1.088E+00	1.088E+00	1.088E+00	1.088E+00	1.088E+00
ELEMENT SPAN	FT	1.584E+00	1.449E+00	1.351E+00	1.160E+00	8.899E-01	5.594E-01	1.908E-01
LEADING EDGE SWEEP	DEG	5.730E-01	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
AERODYNAMIC SWEEP	DEG	1.558E-01	1.758E-01	1.595E-01	1.440E-01	1.356E-01	1.297E-01	1.282E-01
ANGLES OF ATTACK								
GEOMETRIC PITCH	DEG	8.020E-01	1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.000E+00
ZERO LIFT LINE	DEG	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00	-1.238E+00
EFFECTIVE ANGLE	DEG	1.699E+01	1.449E+01	1.285E+01	1.132E+01	9.783E+00	7.860E+00	4.768E+00
WIND ANGLE	DEG	1.768E+01	1.539E+01	1.374E+01	1.253E+01	1.168E+01	1.115E+01	1.090E+01
INFLOW ANGLE	DEG	1.621E+01	1.336E+01	1.160E+01	9.960E+00	8.314E+00	6.266E+00	2.977E+00
INDUCED ANGLE	DEG	-1.356E+00	-1.968E+00	-2.092E+00	-2.544E+00	-3.378E+00	-4.945E+00	-8.014E+00
VELOCITIES								
INDUCED VELOCITY	FT/SEC	1.721E+00	1.789E+00	1.761E+00	1.713E+00	1.508E+00	8.322E-01	-3.257E-01
RELATIVE VELOCITY	FT/SEC	1.310E+02	1.495E+02	1.669E+02	1.824E+02	1.948E+02	2.026E+02	2.054E+02
REYNOLDS NUMBER		9.381E+05	1.040E+06	1.161E+06	1.269E+06	1.355E+06	1.410E+06	1.429E+06
RELATIVE Q	PSF	2.041E+01	2.658E+01	3.313E+01	3.957E+01	4.510E+01	4.881E+01	5.014E+01
AERODYNAMIC COEFFICIENTS								
ELE LIFT COEF		1.265E+00	1.351E+00	1.264E+00	1.155E+00	1.001E+00	7.852E-01	4.906E-01
ELE DRAG COEF		8.088E-02	4.011E-02	2.368E-02	1.784E-02	1.434E-02	1.247E-02	1.000E-02
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
FORCE LOADING								
ELE X FORCE	LBS/FT	-1.612E+03	-2.022E+03	-2.130E+03	-2.009E+03	-1.640E+03	-9.967E+02	-2.163E+02
ELE IND X FORCE	LBS/FT	-2.066E+03	-2.311E+03	-2.345E+03	-2.203E+03	-1.818E+03	-1.165E+03	-3.558E+02
ELE PRO X FORCE	LBS/FT	4.544E+02	2.889E+02	2.141E+02	1.937E+02	1.782E+02	1.685E+02	1.395E+02
ELE Z FORCE	LBS/FT	7.201E+03	9.749E+03	1.140E+04	1.251E+04	1.240E+04	1.057E+04	6.812E+03
ELE IND Z FORCE	LBS/FT	7.070E+03	9.680E+03	1.136E+04	1.247E+04	1.237E+04	1.055E+04	6.805E+03
ELE PRO Z FORCE	LBS/FT	1.314E+02	6.823E+01	4.370E+01	3.383E+01	2.590E+01	1.840E+01	7.213E+00
TORQUE LOADING								
ELE PRO TORQUE	FT-LBS/FT	4.342E+03	3.197E+03	2.667E+03	2.655E+03	2.624E+03	2.602E+03	2.206E+03
ELE IND TORQUE	FT-LBS/FT	-1.975E+04	-2.557E+04	-2.920E+04	-3.019E+04	-2.677E+04	-1.800E+04	-5.628E+03
ELE TORQUE	FT-LBS/FT	-1.540E+04	-2.237E+04	-2.654E+04	-2.753E+04	-2.414E+04	-1.539E+04	-3.421E+03
PITCHING MOMENT LOADING (2D)								
ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
LIFT SLOPE CONVERGENCE								
INPUT LIFT SLOPE	/RAD	4.522E+00	5.388E+00	5.683E+00	5.888E+00	5.913E+00	5.743E+00	5.902E+00
CL/ALPHAE	/RAD	4.330E+00	5.401E+00	5.681E+00	5.886E+00	5.890E+00	5.742E+00	5.902E+00
CIRCULATION PARAMETERS								
CIRC SHAPE PARM		9.405E-01	9.701E-01	9.799E-01	9.906E-01	1.004E+00	1.042E+00	1.167E+00
BOUND CIRCULATION	FT**2/SEC	9.692E+01	1.097E+02	1.148E+02	1.147E+02	1.065E+02	8.655E+01	5.480E+01

```
COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105
-------------------------------------------------------------------------------------------------------------------------------

POINT  8: CARTER 25 *** CMA STALL MODEL *** 05
ROTOR PITCH       (DEG)       .0000
WIND SPEED      (M/SEC)     12.0000
ROTOR AZIMUTH     (DEG)       .0000

ROTOR PERFORMANCE
-----------------

THRUST                      LBS          7.9598E+02
CT ROTOR                                 9.6418E-03
CT WT                                    5.3703E-01

TORQUE                      FT-LBS      -1.6217E+03
CQ ROTOR                                -1.2277E-03
CP WT                                   -3.6086E-01

POWER (ROTOR)               KW          -2.8551E+01
POWER (PROFILE)             KW           5.6907E+00
POWER (INFLOW)              KW          -3.4241E+01
```

RECORD IMAGE OF INPUT FILE

```
ENERTECH 1500 *** CMA STALL MODEL *** 05
3.0       0.0       2.0       0.0       2.0       2.0       0.0
1.0
0.0       0.0       0.0
0.0       0.0       0.0       0.0
0.0       170.0     1.0       1.0
4.0       1.0       15.0      1.0
0.0       0.0       1.0       400.0
3.0       2.0       2.0       6.560
1.3333    12.0      0.0       180.0     30.0      1.0
0.0       0.0       1.0       14.0      15.0      17.0
10.0      1.0       0.25      0.840     5.057760
0.0       0.0       0.0       0.0       0.0       0.0
2.0
0.0       5.057760
0.58333   0.58333
2.0
0.0       6.560
-5.7294   0.0
2.0
0.0       6.560
0.0       0.0
1.0
130.0     0.0       6.560
5.0       4.0       0.25      5.057760  6.560
0.0       0.0       0.0       0.0       0.0       0.0
2.0
5.057760  6.560
0.58333   0.541667
2.0
0.0       6.560
-5.7294   0.0
2.0
0.0       6.560
0.0       0.0
1.0
130.0     0.0       6.560
1.0       3.0
```

```
130.0     NACA 4415 *** VDT 1163  TR 586 *** CMA STALL MODEL
5.0       15.0      5.0       14.0      1.0       2.0
0.15      32.0      1.0       1.0       0.0       2.0       0.5960
          0.0417    0.0830    0.1630    0.3340    0.6540
-6.0      -0.375    -0.175    -0.175    -0.200    -0.200
-4.0      -0.120    0.0100    0.0100    0.0100    0.0100
-2.0      0.1100    0.2100    0.2100    0.2100    0.2100
0.0       0.3300    0.4000    0.4000    0.4000    0.4000
2.0       0.5750    0.6000    0.6000    0.6000    0.6000
4.0       0.7500    0.7800    0.8100    0.8100    0.8100
6.0       0.9100    0.9400    0.9800    1.0000    1.0000
8.0       1.0800    1.0800    1.0800    1.1400    1.1400
9.0       1.1250    1.1250    1.1250    1.1750    1.1900
10.0      1.1900    1.1900    1.2000    1.2000    1.2500
11.0      1.2200    1.2200    1.2200    1.2500    1.3000
12.0      1.2450    1.2450    1.2450    1.2900    1.3400
13.0      1.1850    1.2500    1.2500    1.3050    1.3750
14.0      1.0950    1.2350    1.2600    1.3200    1.3800
15.0      1.0150    1.1900    1.2600    1.3200    1.3900
          0.0417    0.0830    0.1630    0.3340    0.6540
-6.0      0.0460    0.0330    0.0215    0.0160    0.0145
-4.0      0.0380    0.0265    0.0170    0.0130    0.0120
-2.0      0.0340    0.0250    0.0160    0.0125    0.0120
0.0       0.0320    0.0230    0.0150    0.0125    0.0115
2.0       0.0305    0.0230    0.0155    0.0135    0.0115
4.0       0.0305    0.0240    0.0180    0.0155    0.0140
6.0       0.0320    0.0275    0.0220    0.0190    0.0165
8.0       0.0375    0.0335    0.0270    0.0230    0.0210
9.0       0.0415    0.0375    0.0315    0.0255    0.0240
10.0      0.0495    0.0445    0.0390    0.0270    0.0275
11.0      0.0575    0.0520    0.0425    0.0330    0.0355
12.0      0.0716    0.0662    0.0490    0.0475    0.0475
13.0      0.0868    0.0816    0.0650    0.0650    0.0640
14.0      0.1031    0.0981    0.0821    0.0821    0.0811
          0.0
-180.0    0.0
180.0     0.0
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

ENERTECH 1500 *** CMA STALL MODEL *** 05

REFERENCE BLADE SURFACE PANEL COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10	11
1	X	-.022226	-.022230	-.022230	-.022230	-.022229	-.022227	-.022225	-.022221	-.022216	-.022211	-.022204
	Y	.128049	.192344	.256639	.320934	.385229	.449524	.513820	.578115	.642410	.706705	.771000
	Z	-.000453	-.000202	-.000059	.000084	.000227	.000369	.000511	.000653	.000796	.000938	.001080
2	X	.003813	.003814	.003814	.003814	.003814	.003814	.003813	.003813	.003812	.003811	.003810
	Y	.128049	.192344	.256639	.320934	.385229	.449524	.513820	.578115	.642410	.706705	.771000
	Z	.000078	.000035	.000010	-.000014	-.000039	-.000063	-.000088	-.000112	-.000136	-.000161	-.000185
3	X	.066678	.066689	.066691	.066691	.066688	.066682	.066674	.066663	.066649	.066632	.066613
	Y	.128049	.192344	.256639	.320934	.385229	.449524	.513820	.578115	.642410	.706705	.771000
	Z	.001359	.000605	.000176	-.000252	-.000681	-.001108	-.001534	-.001960	-.002387	-.002813	-.003239

I,J		12	13	14	15	16	17
1	X	-.022204	-.021706	-.021256	-.020899	-.020669	-.020590
	Y	.771000	.841765	.905603	.956265	.988792	1.000000
	Z	.001080	.001209	.001321	.001405	.001457	.001474
2	X	.003810	.003724	.003647	.003586	.003546	.003533
	Y	.771000	.841765	.905603	.956265	.988792	1.000000
	Z	-.000185	-.000208	-.000227	-.000241	-.000250	-.000253
3	X	.066613	.065119	.063769	.062697	.062008	.061770
	Y	.771000	.841765	.905603	.956265	.988792	1.000000
	Z	-.003239	-.003628	-.003962	-.004214	-.004370	-.004423

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

ENERTECH 1500 *** CMA STALL MODEL *** 05

REFERENCE BLADE COLLOCATION POINT COORDINATES

I,J		1	2	3	4	5	6	7	8	9	10
1	X	-.013203	-.013204	-.013204	-.013203	-.013203	-.013201	-.013199	-.013197	-.013194	-.013190
	Y	.160196	.224491	.288787	.353082	.417377	.481672	.545967	.610262	.674557	.738852
	Z	-.000162	-.000077	.000008	.000092	.000177	.000261	.000346	.000430	.000514	.000599
2	X	.029017	.029018	.029019	.029018	.029016	.029013	.029009	.029003	.028997	.028989
	Y	.160196	.224491	.288787	.353082	.417377	.481672	.545967	.610262	.674557	.738852
	Z	.000357	.000170	-.000017	-.000203	-.000389	-.000574	-.000760	-.000945	-.001131	-.001316

I,J		11	12	13	14	15
1	X	-.013040	-.012759	-.012519	-.012345	-.012253
	Y	.806382	.873684	.930934	.972528	.994396
	Z	.000680	.000752	.000810	.000850	.000870
2	X	.028660	.028041	.027514	.027131	.026929
	Y	.806382	.873684	.930934	.972528	.994396
	Z	-.001495	-.001652	-.001779	-.001868	-.001913

```
COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105
-------------------------------------------------------------------------------------------------------------------------

POINT  7: ENERTECH 1500 *** CMA STALL MODEL *** 05
ROTOR PITCH       (DEG)       .0000
WIND SPEED      (M/SEC)     10.0000
ROTOR AZIMUTH     (DEG)       .0000

WORKING STATE AND WAKE RATE PARAMETERS
--------------------------------------

WINDMILL BRAKE STATE

  REFERRED VELOCITIES,
    AXIAL ONSET   2.90268E+00
    INDUCED      -3.99491E-01

  REFERENCE WAKE PARAMETERS,
    CTRW          1.87344E-02
    AKZO         -1.36874E-01
    ALAMO         5.47495E-01
    AKRO         -4.00000E+00

  MODIFIED WAKE PARAMETERS,
    AKZW          2.80934E-01
    AKZI         -3.10553E-02
    AKZ           2.49878E-01
    AKR           2.19105E+00
    A             1.22000E+00
    AKB           1.82561E+00
```

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 7: ENERTECH 1500 *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 10.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

GEOMETRY

ELEMENT NUMBER		1	2	3	4	5	6	7	8
ELE CENTER STATION		1.602E-01	2.245E-01	2.888E-01	3.531E-01	4.174E-01	4.817E-01	5.460E-01	6.103E-01
CHORD	FT	5.833E-01	5.833E-01	5.833E-01	5.833E-01	5.833E-01	5.833E-01	5.833E-01	5.833E-01
ELEMENT SPAN	FT	4.218E-01	4.218E-01	4.218E-01	4.218E-01	4.218E-01	4.218E-01	4.218E-01	4.218E-01
LEADING EDGE SWEEP	DEG	.000E+00	.000E+00	1.978E-02	.000E+00	.000E+00	1.978E-02	.000E+00	.000E+00
AERODYNAMIC SWEEP	DEG	4.120E+00	3.600E+00	3.251E+00	2.904E+00	2.603E+00	2.348E+00	2.140E+00	1.960E+00
ANGLES OF ATTACK									
GEOMETRIC PITCH	DEG	-4.812E+00	-4.443E+00	-4.075E+00	-3.706E+00	-3.338E+00	-2.970E+00	-2.601E+00	-2.233E+00
ZERO LIFT LINE	DEG	-4.107E+00	-4.107E+00	-4.107E+00	-4.105E+00	-4.104E+00	-4.102E+00	-4.100E+00	-4.098E+00
EFFECTIVE ANGLE	DEG	5.284E+01	4.649E+01	3.983E+01	3.501E+01	3.132E+01	2.832E+01	2.536E+01	2.288E+01
WIND ANGLE	DEG	6.031E+01	5.137E+01	4.421E+01	3.851E+01	3.394E+01	3.025E+01	2.723E+01	2.472E+01
INFLOW ANGLE	DEG	5.856E+01	5.071E+01	4.288E+01	3.721E+01	3.282E+01	2.921E+01	2.568E+01	2.267E+01
INDUCED ANGLE	DEG	-2.321E+00	9.238E-01	-6.884E-01	-7.851E-01	-6.653E-01	-6.378E-01	-1.291E+00	-1.878E+00
VELOCITIES									
INDUCED VELOCITY	FT/SEC	8.619E-01	9.886E-01	1.150E+00	1.205E+00	1.225E+00	1.283E+00	1.414E+00	1.570E+00
RELATIVE VELOCITY	FT/SEC	3.762E+01	4.304E+01	4.767E+01	5.336E+01	5.952E+01	6.597E+01	7.237E+01	7.898E+01
REYNOLDS NUMBER		1.403E+05	1.606E+05	1.778E+05	1.990E+05	2.220E+05	2.461E+05	2.699E+05	2.946E+05
RELATIVE Q	PSF	1.683E+00	2.203E+00	2.702E+00	3.385E+00	4.212E+00	5.175E+00	6.227E+00	7.417E+00
AERODYNAMIC COEFFICIENTS									
ELE LIFT COEF		5.879E-01	6.664E-01	8.259E-01	9.068E-01	9.487E-01	9.952E-01	1.059E+00	1.135E+00
ELE DRAG COEF		6.683E-01	5.865E-01	5.696E-01	5.123E-01	3.710E-01	2.765E-01	2.009E-01	1.501E-01
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
FORCE LOADING									
ELE X FORCE	LBS/FT	-3.000E+00	-7.981E+00	-9.814E+00	-1.193E+01	-2.141E+01	-3.174E+01	-4.344E+01	-5.569E+01
ELE IND X FORCE	LBS/FT	-1.772E+01	-2.852E+01	-3.812E+01	-4.660E+01	-5.437E+01	-6.309E+01	-7.174E+01	-8.148E+01
ELE PRO X FORCE	LBS/FT	1.472E+01	2.054E+01	2.831E+01	3.467E+01	3.297E+01	3.135E+01	2.830E+01	2.578E+01
ELE Z FORCE	LBS/FT	3.708E+01	4.844E+01	6.734E+01	8.769E+01	1.056E+02	1.304E+02	1.628E+02	2.059E+02
ELE IND Z FORCE	LBS/FT	1.299E+01	2.334E+01	4.105E+01	6.137E+01	8.430E+01	1.128E+02	1.492E+02	1.951E+02
ELE PRO Z FORCE	LBS/FT	2.409E+01	2.510E+01	2.629E+01	2.633E+01	2.126E+01	1.753E+01	1.361E+01	1.077E+01
TORQUE LOADING									
ELE PRO TORQUE	FT-LBS/FT	1.547E+01	3.025E+01	5.363E+01	8.030E+01	9.026E+01	9.906E+01	1.013E+02	1.032E+02
ELE IND TORQUE	FT-LBS/FT	-1.862E+01	-4.200E+01	-7.222E+01	-1.079E+02	-1.489E+02	-1.993E+02	-2.569E+02	-3.262E+02
ELE TORQUE	FT-LBS/FT	-3.152E+00	-1.175E+01	-1.859E+01	-2.764E+01	-5.861E+01	-1.003E+02	-1.556E+02	-2.230E+02
PITCHING MOMENT LOADING (2D)									
ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
LIFT SLOPE CONVERGENCE									
INPUT LIFT SLOPE	/RAD	7.349E-01	9.375E-01	1.301E+00	1.579E+00	1.828E+00	2.117E+00	2.570E+00	3.083E+00
CL/ALPHAE	/RAD	7.377E-01	9.189E-01	1.289E+00	1.580E+00	1.825E+00	2.098E+00	2.472E+00	2.921E+00
CIRCULATION PARAMETERS									
CIRC SHAPE PARM		8.377E-01	8.357E-01	8.490E-01	8.568E-01	8.632E-01	8.707E-01	8.859E-01	9.028E-01
BOUND CIRCULATION	FT**2/SEC	6.415E+00	8.524E+00	1.157E+01	1.409E+01	1.648E+01	1.931E+01	2.322E+01	2.760E+01

COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105

POINT 7: ENERTECH 1500 *** CMA STALL MODEL *** 05
ROTOR PITCH (DEG) .0000
WIND SPEED (M/SEC) 10.0000
ROTOR AZIMUTH (DEG) .0000

BLADE 1 ELEMENT CHARACTERISTICS

		9	10	11	12	13	14	15
GEOMETRY								
ELEMENT NUMBER		9	10	11	12	13	14	15
ELE CENTER STATION		6.746E-01	7.389E-01	8.064E-01	8.737E-01	9.309E-01	9.725E-01	9.944E-01
CHORD	FT	5.833E-01	5.833E-01	5.769E-01	5.646E-01	5.542E-01	5.467E-01	5.427E-01
ELEMENT SPAN	FT	4.218E-01	4.218E-01	4.642E-01	4.188E-01	3.323E-01	2.134E-01	7.352E-02
LEADING EDGE SWEEP	DEG	1.978E-02	.000E+00	3.976E-01	3.971E-01	3.971E-01	3.971E-01	3.976E-01
AERODYNAMIC SWEEP	DEG	1.809E+00	1.674E+00	1.534E+00	1.400E+00	1.309E+00	1.261E+00	1.247E+00
ANGLES OF ATTACK								
GEOMETRIC PITCH	DEG	-1.865E+00	-1.496E+00	-1.109E+00	-7.237E-01	-3.957E-01	-1.574E-01	-3.211E-02
ZERO LIFT LINE	DEG	-4.096E+00	-4.096E+00	-4.096E+00	-4.096E+00	-4.096E+00	-4.096E+00	-4.096E+00
EFFECTIVE ANGLE	DEG	2.051E+01	1.874E+01	1.736E+01	1.580E+01	1.346E+01	1.007E+01	6.336E+00
WIND ANGLE	DEG	2.261E+01	2.082E+01	1.921E+01	1.783E+01	1.679E+01	1.611E+01	1.578E+01
INFLOW ANGLE	DEG	1.979E+01	1.614E+01	1.437E+01	1.243E+01	9.756E+00	6.130E+00	2.272E+00
INDUCED ANGLE	DEG	-2.734E+00	-4.678E+00	-4.834E+00	-5.399E+00	-7.036E+00	-9.983E+00	-1.350E+01
VELOCITIES								
INDUCED VELOCITY	FT/SEC	1.715E+00	1.808E+00	1.846E+00	1.812E+00	1.497E+00	4.092E-01	-1.043E+00
RELATIVE VELOCITY	FT/SEC	8.559E+01	9.243E+01	9.976E+01	1.069E+02	1.122E+02	1.149E+02	1.153E+02
REYNOLDS NUMBER		3.192E+05	3.447E+05	3.680E+05	3.859E+05	3.978E+05	4.015E+05	4.000E+05
RELATIVE Q	PSF	8.710E+00	1.016E+01	1.183E+01	1.358E+01	1.498E+01	1.569E+01	1.580E+01
AERODYNAMIC COEFFICIENTS								
ELE LIFT COEF		1.238E+00	1.321E+00	1.315E+00	1.285E+00	1.189E+00	9.969E-01	6.249E-01
ELE DRAG COEF		1.118E-01	8.905E-02	6.935E-02	4.327E-02	2.587E-02	1.842E-02	1.334E-02
ELE MOMENT COEF		.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
FORCE LOADING								
ELE X FORCE	LBS/FT	-6.863E+01	-6.356E+01	-6.840E+01	-7.164E+01	-6.351E+01	-4.381E+01	-1.497E+01
ELE IND X FORCE	LBS/FT	-9.163E+01	-8.537E+01	-8.813E+01	-8.559E+01	-7.262E+01	-5.057E+01	-1.989E+01
ELE PRO X FORCE	LBS/FT	2.300E+01	2.181E+01	1.974E+01	1.395E+01	9.109E+00	6.759E+00	4.915E+00
ELE Z FORCE	LBS/FT	2.629E+02	3.295E+02	3.789E+02	4.171E+02	4.202E+02	3.673E+02	2.313E+02
ELE IND Z FORCE	LBS/FT	2.546E+02	3.232E+02	3.739E+02	4.140E+02	4.186E+02	3.666E+02	2.311E+02
ELE PRO Z FORCE	LBS/FT	8.277E+00	6.312E+00	5.058E+00	3.074E+00	1.566E+00	7.258E-01	1.950E-01
TORQUE LOADING								
ELE PRO TORQUE	FT-LBS/FT	1.018E+02	1.057E+02	1.044E+02	7.995E+01	5.563E+01	4.312E+01	3.206E+01
ELE IND TORQUE	FT-LBS/FT	-4.055E+02	-4.138E+02	-4.662E+02	-4.906E+02	-4.435E+02	-3.226E+02	-1.297E+02
ELE TORQUE	FT-LBS/FT	-3.037E+02	-3.081E+02	-3.618E+02	-4.106E+02	-3.878E+02	-2.795E+02	-9.767E+01
PITCHING MOMENT LOADING (2D)								
ELE PITCH MOMENT	FT-LBS/FT	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
LIFT SLOPE CONVERGENCE								
INPUT LIFT SLOPE	/RAD	3.685E+00	4.143E+00	4.441E+00	4.731E+00	5.166E+00	5.687E+00	5.670E+00
CL/ALPHAE	/RAD	3.532E+00	4.112E+00	4.408E+00	4.722E+00	5.109E+00	5.702E+00	5.663E+00
CIRCULATION PARAMETERS								
CIRC SHAPE PARM		9.245E-01	9.406E-01	9.513E-01	9.666E-01	1.004E+00	1.093E+00	1.212E+00
BOUND CIRCULATION	FT**2/SEC	3.222E+01	3.587E+01	3.812E+01	3.886E+01	3.738E+01	3.121E+01	1.957E+01

```
COMPUTATIONAL METHODOLOGY ASSOCIATES *** LSWT VERSION 1.00 RELEASED MAY 1986 *** SERI IL-5-05105
-------------------------------------------------------------------------------------------------------------------------------

POINT  7: ENERTECH 1500 *** CMA STALL MODEL *** 05
ROTOR PITCH       (DEG)       .0000
WIND SPEED      (M/SEC)     10.0000
ROTOR AZIMUTH     (DEG)       .0000

ROTOR PERFORMANCE
-----------------

THRUST                      LBS          8.3094E+01
CT ROTOR                                 1.8951E-02
CT WT                                    4.8024E-01

TORQUE                      FT-LBS      -7.2929E+01
CQ ROTOR                                -2.5355E-03
CP WT                                   -2.2871E-01

POWER (ROTOR)               KW          -1.7603E+00
POWER (PROFILE)             KW           7.4273E-01
POWER (INFLOW)              KW          -2.5030E+00
```

Document Control Page	1. SERI Report No. SERI/STR-217-3163	2. NTIS Accession No.	3. Recipient's Accession No.

4. Title and Subtitle	5. Publication Date
Lifting Surface Performance Analysis for Horizontal Axis Wind Turbines	June 1987
	6.
7. Author(s) D. Kocurek	8. Performing Organization Rept. No.
9. Performing Organization Name and Address Computational Methodology Associates 2900 Steve Drive Hurst, Texas 76054	10. Project/Task/Work Unit No. 4845.10
	11. Contract (C) or Grant (G) No. (C) IL-5-05105-1 (G)
12. Sponsoring Organization Name and Address Solar Energy Research Institute A Division of Midwest Research Institute 1617 Cole Blvd. Golden, Colorado 80401-3393	13. Type of Report & Period Covered Technical Report
	14.

15. Supplementary Notes

Technical Monitor: James Tangler

16. Abstract (Limit: 200 words) This report describes how numerical lifting-surface theory is applied to the calculation of a horizontal-axis wind turbine's aerodynamic characteristics and performance. The report also describes how such an application is implemented as a computer program. The method evolved from rotary-wing and helicopter applications and features a detailed, prescribed wake. The wake model extends from a hovering-rotor experimental generalization to include the effect of the windmill brake state on the radial and axial displacement rates of the trailing vortex system. Performance calculations are made by coupling the lifting-surface circulation solution to a blade-element analysis that incorporates two-dimensional airfoil characteristics as functions of angle of attack and Reynolds number. Several analytical stall models are also provided to extend the airfoil characteristics beyond the limits of available data. Although this work focuses on the steady-performance problem, the method includes ways to investigate the effects of wind-shear profile, tower shadow, and off-axis shaft alignment. Correlating the method to measured wind-turbine performance, and comparing it to blade-element momentum theory calculations, validate and highlight the extreme sensitivity of predictions to the quality of early post-stall airfoil bahavior.

17. Document Analysis

a. Descriptors Wind Energy ; Wind Turbine ; Performance Analysis ; Simulation ; Numerical Theory ; Aerodynamic Characteristics

b. Identifiers/Open-Ended Terms

c. UC Categories

18. Availability Statement	19. No. of Pages
National Technical Information Service U.S. Department of Commerce 5285 Port Royal Road Springfield, Virginia 22161	319
	20. Price A14

Form No. 0069 (3-25-82)