DOE/NASA/0005-2
NASA CR-174921

NASA-CR-174921
19850025250

Aerodynamic Analysis of a Horizontal Axis Wind Turbine by Use of Helical Vortex Theory

Volume II: Computer Program Users Manual

T G Keith, Jr , A A. Afjeh, D R Jeng,
and J A. White
The University of Toledo

April 1985

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Under Grant NCC 3-5

for
U.S. DEPARTMENT OF ENERGY
Conservation and Renewable Energy
Wind Energy Technology Division

DISCLAIMER

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

NTIS price codes[1]
Printed copy A05
Microfiche copy A01

[1]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 issues 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

DOE/NASA/0005-2
NASA CR-174921

Aerodynamic Analysis of a Horizontal Axis Wind Turbine by Use of Helical Vortex Theory

Volume II: Computer Program Users Manual

T.G Keith, Jr., A.A Afjeh, D R Jeng,
and J.A. White
The University of Toledo
Toledo, Ohio

April 1985

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Cleveland, Ohio 44135
Under Grant NCC 3-5

for
U.S. DEPARTMENT OF ENERGY
Conservation and Renewable Energy
Wind Energy Technology Division
Washington, D.C. 20545
Under Interagency Agreement DE-AI01-76ET20320

N85-33563#

CONTENTS

INTRODUCTION

Recently a report [1] was prepared describing the theoretical underpinnings of a vortex wake method of analysis for prediction of the aerodynamic performance of horizontal axis wind turbines. By use of this method, rotors having any number of blades, which may be arbitarily shaped and twisted, can be analysed. Furthermore, it has been found that a computer code entitled VORTEX which implements the vortex wake method of analysis can obtain answers to problems of interest in computing times that are comparable to those obtained from more elementary methods e.g. the blade element-momentum theory used in the well known PROP computer code [2] or in the WIND II code [3].

In the vortex wake method, the induced velocity is directly obtained by integration of the Biot-Savart law. To implement this integration, it was assumed that a discrete number of vortex filaments trail from the rotor blade. These filaments extend infinitely far downstream and have a constant diameter helical shape. It was also assumed that the entire helical vortex system produced by the rotating blades travels downstream with a constant velocity equal to the value of the rotor disk. Lifting line theory was used to represent the blades and aerodynamic effects were incorporated by use of empirical airfoil lift and drag curves.

Aerodynamic performance predictions using the vortex wake method of analysis were compared to experimental data for four different rotors in [4]. In general, favorable agreement between measured and predicted rotor power was obtained. The method has also been recently extended to include performance prediction of tip-controlled wind turbines [5], [6] and [7]. Moreover, a vortex wake model was found to provide better simulation of an experimentally observed post-peak power plateau than did a blade element-momentum model [8].

A significant drawback of the calculation procedure needed to implement the vortex wake method of analysis is that it is unavoidably quite involved. The reasons for this complexity are better understood once the overall procedure has been outlined. Essentials of the procedure were given in [1], however, certain important numerical details were omitted in order to avoid complicating the presentation of the theory. In addition, no discussion of the computer code that evolved from that study was presented. As a consequence, the current report is written with two principal objectives in mind:

(1) to supply missing details in the computational procedure, and

(2) to describe the computer program VORTEX developed to implement the vortex wake theory.

In order to accomplish these objectives, both the numerical procedure and the computer program will be fully discussed in the following. Details of the integration and interpolation schemes that were used will be presented along with a method for treating numerical singularities that occur in some of the governing integral expressions. All program variables and subroutines will be defined and input/output parameters will be described. Finally, program implementation will be illustrated by the presentation of an example problem.

THE NUMERICAL PROCEDURE

For calculation purposes each blade of the wind turbine is thought to be divided into M-1 spanwise sections. These subdivisions establish M calculation positions along each blade. Although the subdividing is arbitrary, it must be understood that a large number of subdivisions can lead to excessive computational times without necessarily improving overall accuracy. In most problems solved to date, 9 subdivisions were found to provide good balance between accuracy and computer run times. It should also be mentioned that the subdivisions need not be of equal size e.g., more sections may be located near the blade tip or in regions of high gradients. To simplify the computations, it is presumed that each of the N rotor blades of the wind turbine has been subdivided into the same number and distribution of parts.

In the vortex wake method of analysis, it is assumed that each rotor blade can be represented by a single bound vortex i.e., the lifting line. And because no vortex line may end abruptly within the fluid, it is assumed that there is a system of vortices that trails each rotor blade. Accordingly, it is appropriate to think of a vortex filament emanating from end points of a blade subdivision. The collection of these trailing vortex filaments form a system of vortex sheets having a helical geometry due to the blade rotation. These vortex sheets extend infinitely far downstream of the rotor and are assumed to have a constant diameter and constant pitch; this is known as the rigid wake assumption. In point of fact, the wake is not rigid but expands radially in the downstream direction.

In performing the calculations, the influence that each trailing vortex filament has on the flowfield in the vicinity of each bound vortex must be determined. The combined influence is called the induced velocity or the downwash. The downwash can be found by direct integration of , the Biot-Savart law [9]. This integration is made somewhat difficult by the fact that it must be performed over the entire length of every helical vortex filament. The downwash is subsequently used to calculate the induced angle of attack distribution which in turn may be used to evaluate various performance factors, e.g., rotor power, rotor thrust, etc. To accomplish all of this, the circulation distribution along the blade, $\Gamma(\xi)$, where ξ is the nondimensional spanwise position dimension, must be known. As will be shown in the following, the particular form of the distribution evolves from the calculation procedure having initially been assumed in the form of a truncated Fourier sine series:

$$\Gamma = \sum_{m=1}^{M} A_m \sin\left[(m\pi)\ \frac{\xi - \xi_{hub}}{1 - \xi_{hub}}\right] \tag{1}$$

It should be noticed that this distribution has been constructed so that the circulation at both the blade tip ($\xi=1$) and the blade hub ($\xi=\xi_{hub}$) vanishes. The lift and drag coefficients for each blade section must also be known. Generally, this information is supplied in the form of curve-fitted wind tunnel airfoil data.

In order to have an understanding of the calculation procedure, the following step-by-step outline and description is presented.

Step 1 Select M stations along the blade span.

Each calculation site on the blade is designated by a ξ' value. As mentioned, generally nine (9) stations have been used in the computations. For the numerical integration, a separate nodal system of blade positions is required. These locations will be designated as ξ. Obviously, there are many more ξ locations than ξ' locations.

Step 2 Determine the circulation distribution along each blade (first pass).

In the calculation procedure, the circulation distribution was determined in one of two ways depending on whether it was the first pass through the calculation procedure. In the first pass, it was found helpful to write the effective angle of attack for each blade section, $(\alpha_e)_m$, as a function of sectional lift coefficient, $(C_L)_m$. This expression when combined with: (a) the Kutta-Joukowski Theorem [9], (b) an expression for the induced angle of attack and (c) the circulation distribution produces a rather complicated expression {see equation (2-67) in [1]} which can be written in compact form as

$$\sum_{m=1}^{M} f(\xi')A_m = g(\xi') \qquad (2)$$

In this equation, both f and g are functions that involve many parameters and the A_m are the coefficients of the circulation distribution in equation (1). By applying equation (2) at each ξ' location, a set of M equations in M unknowns is obtained. Solution of that set gives the A_m which permits the circulation to be determined.

Step 3 Calculation of the induced angle of attack.

At each station, ξ', the induced angle of attack, $(\alpha_i)_m$, can be computed from the circulation distribution.

Step 4 Calculation of the effective angle of attack.

From the relation between geometric, effective and induced angles of attack, new values of $(\alpha_e)_m$ can be evaluated.

Step 5 Calculations of the lift coefficient.

Using two-dimensional airfoil data and the values of the effective angle of attack, new values of $(C_L)_m$ can be determined. If these agree with the previous $(C_L)_m$, the iteration procedure is terminated. If they do not agree, the process continues to the next step.

Step 6 Determination of the circulation distribution.

The Kutta-Joukowski theorem applied to each blade section may be written

$$\Gamma_m = \frac{1}{2} (CWC_L)_m \tag{3}$$

Values of $(C_L)_m$ from step 5 permit the Γ_m to be determined. In turn, the A_m in equation (1) may be found from the following matrix equation

$$[C] \{A_m\} = \{\Gamma\} \tag{4}$$

where

$$C_{ij} = \sin \left[(j\pi) \left(\frac{\xi'_i - \xi_{hub}}{1 - \xi_{hub}} \right) \right]$$

Once this calculation is completed, the program is directed to return to step 3.

NUMERICAL ANALYSIS

Because the computer program must perform interpolations and integrations, solve systems of equations and resolve numerical singularities that arise in certain integrals, it is appropriate that this numerical work be described.

Interpolation

In the program, a cubic spline interpolation technique [10] was used in which the second derivative at each end of the interpolated data set is assumed to be a linear extrapolation of the value at the two adjacent points. This interpolation was found to be necessary to calculate the induction factor in the neighborhood of the singularity (see singularity treatment below).

If a cubic polynomial is defined as

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

then between points x_k and x_{k+1}, the second derivative has the value

$$f''(X) = f''_k \left(\frac{X_{k+1} - X}{\delta_k} \right) + f''_{k+1} \left(\frac{X - X_k}{\delta_k} \right) \tag{5}$$

where $\delta_k = X_{k+1} - X_k$. Integrating equation (5) twice yields

$$f(X) = f''_k \left[\frac{(X_{k+1} - X)^3}{6\ \delta_k} \right] + f''_{k+1} \left[\frac{(X - X_k)^3}{6\ \delta_k} \right] + C_1 x + C_2 \tag{6}$$

The integration constants C_1 and C_2 are obtained from the fact that f(X) passes through X_k and X_{k+1}. Thus,

$$C_1 = \frac{(f_{k+1} - f_k)}{\delta_k} - \frac{(f''_{k+1} - f''_k)\delta_k}{6}$$

$$C_2 = \frac{(f_k X_{k+1} - f_{k+1} X_k)}{\delta_k} - \frac{(f''_k X_{k+1} - f''_{k+1} X_k)\delta_k}{6}$$

Inserting these into equation (6) produces

$$f(X) = \frac{f''_k (X_{k+1} - X)^3}{6\ \delta_k} + \frac{f''_{k+1} (X - X_k)^3}{6\ \delta_k} + (X_{k+1} - X)\left(\frac{f_k}{\delta_k} - \frac{f''_k \delta_k}{6}\right)$$

$$+ (X - X_k)\left(\frac{f_{k+1}}{\delta_k} - \frac{f''_{k+1}\delta_k}{6}\right) \qquad (7)$$

In this equation, only the second derivatives f''_k and f''_{k+1} are unknown. However, the value of the second derivative at all points can be obtained by equating the slope of two neighboring subdomains at both ends of the interpolated data set. This results in m-2 equations for m points. Therefore, two additional equations are required. They may be obtained by linearly extrapolating the value of the second derivative at the two points adjacent to both ends of the interpolated domain i.e., at points 1 and m. This results in the following expression for each internal point:

$$\frac{f''_{k-1}\,\delta_{k-1}}{6} + f''_k\left(\frac{\delta_{k-1} + \delta_k}{3}\right) + \frac{f''_{k+1}\,\delta_k}{6} = \frac{f_{k+1} - f_k}{\delta_k} - \frac{f_k - f_{k-1}}{\delta_{k-1}} \tag{8}$$

and the following pair at the end points

$$-\frac{f''_1}{\delta_1} + f''_2\left(\frac{1}{\delta_1} + \frac{1}{\delta_2}\right) - \frac{f''_3}{\delta_2} = 0 \tag{9}$$

$$-\frac{f''_{m-2}}{\delta_{m-2}} + f''_{m-1}\left(\frac{1}{\delta_{m-2}} + \frac{1}{\delta_{m-1}}\right) - \frac{f''_m}{\delta_{m-1}} = 0 \tag{10}$$

This system of linear equations can be solved for f''_k, $k = 1,\ldots.,m$. Thus, the interpolating function is completely determined as

$$f(X) = A_{1,k}(X_{k+1} - X)^3 + A_{2,k}(X - X_k)^3 + A_{3,k}(X_{k+1} - X) + A_{4,k}(X - X_k) \tag{11}$$

where the $A_{i,k}$ are the coefficients of the x values as displayed in equation (7).

Integration

The definite integrals that occur in the analysis are handled by making use of the spline interpolation formula, equation (11) above. The integration formula that is produced is

$$\int f(X)dX = \frac{A_{1,k}}{4}(X_{k+1} - X)^4 + \frac{A_{2,k}}{4}(X - X_k)^4 +$$

$$\frac{A_{3,k}}{2}(X_{k+1} - X)^2 + \frac{A_{4,k}}{2}(X - X_k)^2 \tag{12}$$

The semi-infinite integral that occurs in the equation for the induction factor was determined by using a 24 point Gauss-Laguerre formula, i.e.,

$$\int_0^\infty f(X)e^{-X}dX = \sum_{i=1}^{N} f(X_i)w_i \tag{13}$$

The weighing factors, w_i, can be found in the literature, [11].

Solution of the governing system of equations:

The linear system of equations, equation (4), for the circulation coefficients is solved simultaneously by employing the Gauss elimination method. Because this method is rather elementary, it is not presented here. Those unfamiliar with the technique should consult a standard numerical analysis reference, e.g., [12].

Treatment of the numerical singularities.

In [1], it is shown that the total induced velocity at any point ξ' on the blade lifting line due to all trailing helical vortices from all blades is

$$w_n(\xi') = \int_{\xi_{hub}}^{\xi_{tip}} \frac{\frac{d\Gamma}{d\xi}d\xi}{4\pi R} \sum_{k=1}^{N} \int_{0}^{\infty} \frac{N_1\xi' + N_2\lambda_o}{D_1^{3/2}\, D_2^{1/2}}\, d\theta \tag{14}$$

where

$$N_1 \equiv \xi[\xi - \xi'\cos(\theta + \theta_k)]$$

$$N_2 \equiv [-\xi' + \xi\cos(\theta + \theta_k) + \theta\,\xi\sin(\theta + \theta_k)]\,\frac{h}{R}$$

$$D_1 \equiv \xi^2 + \xi'^2 - 2\xi\xi'\cos(\theta + \theta_k) + \frac{h^2}{R^2}\theta^2$$

$$D_2 \equiv \lambda_o^2 + \xi'^2$$

$$h \equiv \frac{v_o - w_n\cos\phi'}{\Omega + \dfrac{w_n\sin\phi'}{r}}$$

Careful examination of equation (14) reveals that a numerical problem can develop. In particular, if the parameter D_1 becomes zero, as it can when $\theta = \theta_k = 0$ and $\xi = \xi'$ occur simultaneously, the value of the induced velocity at that point on the blade becomes infinite. The presence of these points of singularity prevents a direct solution of equation (14) in the neighborhood of the points.

To overcome this numerical difficulty, a theory developed by Moriya [13] was utilized. This theory is based on the fact that if the differential normal induced velocity [the differential form of equation (14)] was written for a straight trailing vortex (as opposed to the actual helical vortex), this quantity would become unbounded in exactly the same manner as the quantity for a helical vortex. Physically this results from the fact that the major contributing factor which causes equation (14) to become unbounded is due to the segment of the trailing vortex that is very close to ξ' i.e., very close to the trailing edge at ξ'. Clearly, it is assumed that curvature effects of the helical vortex in the vicinity of the singularities are not important. Figure 1 is a graphical representation of this concept. Since the differential normal induced velocity for both the helical vortex [say $(dw_n)_H$] and the straight vortex [say $(dw_n)_S$] approach infinity at the same rate , then their ratio, which Moriya called the induction factor I, i.e.,

$$I = \frac{(dW_n)_H}{(dW_n)_S} \tag{15}$$

tends to unity at each singularity.

Combining equations (14) and (15) along with an expression for $(dW_n)_S$, [9], yields

$$W_n = \int_{\xi_{hub}}^{\xi_{tip}} \frac{\frac{d\Gamma}{d\xi}d\xi}{4\pi R} \frac{I}{(\xi - \xi')} d\xi \tag{16}$$

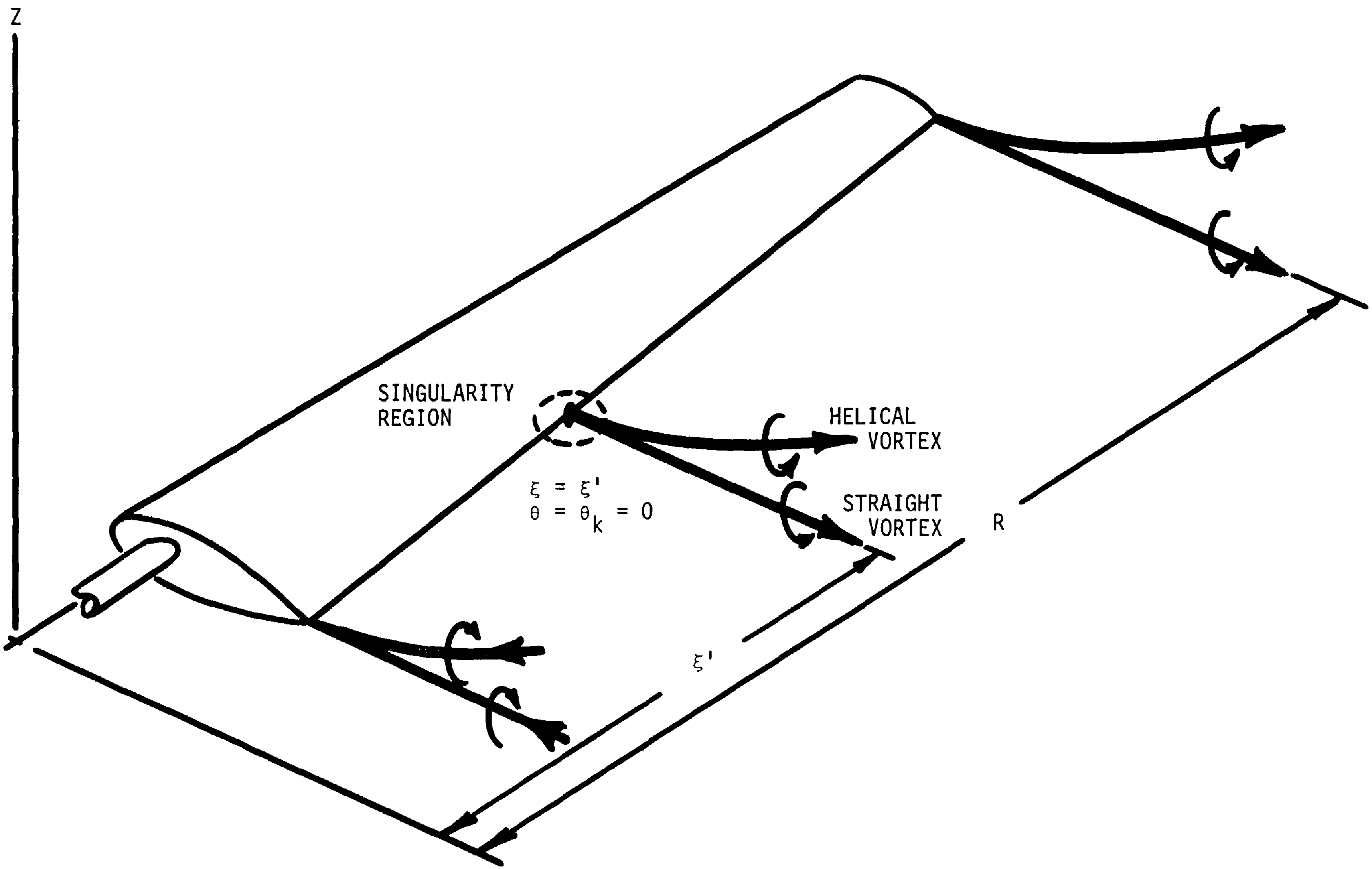

FIG. 1 MORIYA'S THEORY OF SINGULARITIES

where

$$I = I(\xi, \xi', \lambda_0)$$

$$= (\xi - \xi') \sum_{k=1}^{N} \int_0^{\infty} \frac{N_1 \xi' + N_2 \lambda_0}{D_1^{3/2} \; D_2^{1/2}} d\theta \tag{17}$$

Experience with the numerical integration of the induction factor using an integration interval spacing equal to ten times the number of blade segments minus one, times the blade radius revealed a problem. It has been found that for small values of the wind-tip speed ration, λ_0, the numerical value of I tends to oscillate both slightly before and slightly after the location of the singular point. This behavior is diagramed in Fig. 2a. To remove this oscillation in a sensible manner, it was decided to integrate equation (17) to within a small distance of the singularity (denoted by I = 1) and then to use the last four I values to fit a cubic spline through the point I = 1 as diagrammed in Fig. 2b. The distance from the singularity needs to be large enough so not to contain any of the unwanted oscillation yet not so large as to result in a loss of accuracy. To be sure, the selection of the distances on either side of the singularity requires some experimentation.

It should be noted that Moriya did not report encountering any oscillation of the type found in this study. This is not surprising since his work concentrated at much larger values of λ_0 where no such oscillation has been observed.

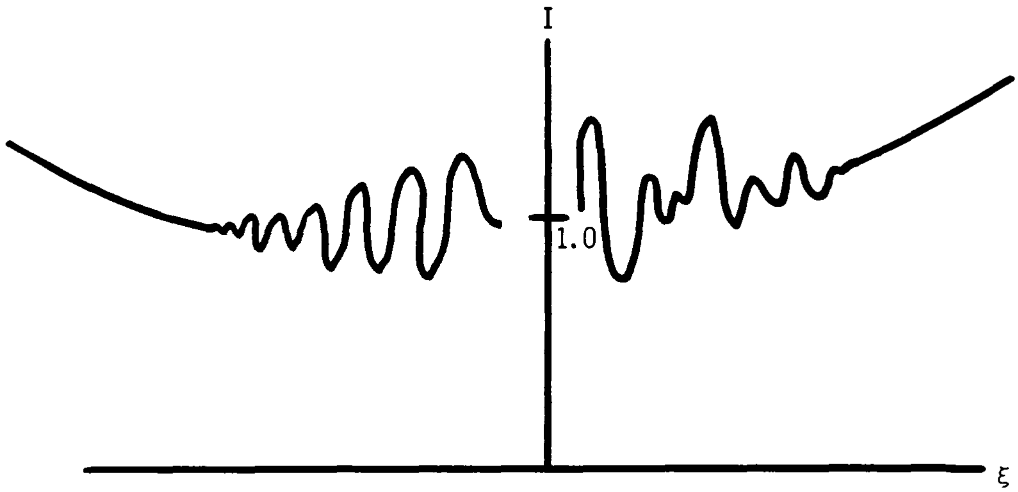

FIG. 2a OSCILLATION IN THE INDUCTION FACTOR INTEGRATION

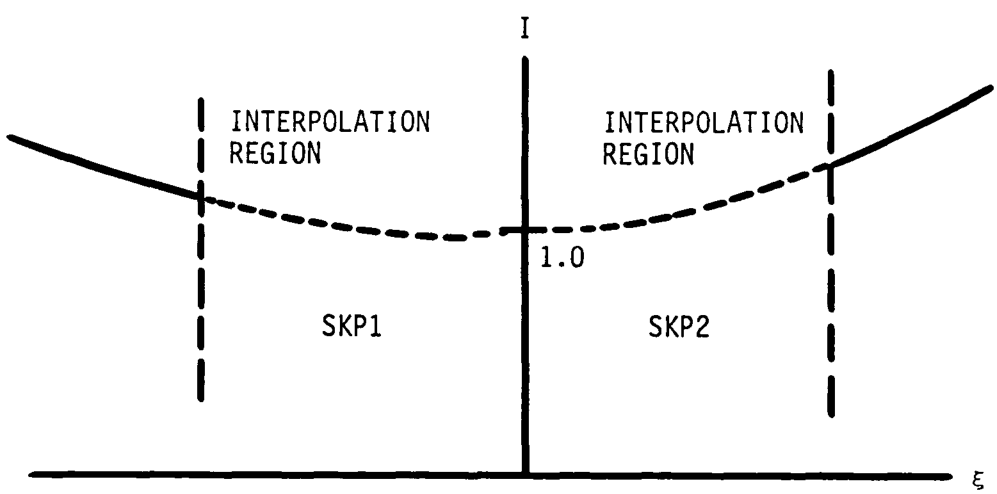

FIG. 2b REMOVAL OF OSCILLATION IN THE INDUCTION FACTOR INTEGRATION

THE COMPUTER PROGRAM

A computer program to implement the numerical procedure described in the previous section has been written in FORTRAN IV. Figure 3 displays a flowchart of that program. It can be seen that the program is divided into five primary subprograms. These subprograms in turn call 11 other secondary subprograms.

In the following each of the primary subprograms will be described in order of their appearance in the program. This description will consist of: (a) a brief outline statement of the function of the subprogram, (b) a listing of the subprogram arguments and local variables, (c) a list of all secondary subprograms called and finally (d) a collection of any important notes relevant to the particular subprogram.

A similar description of the secondary subprograms, listed alphabetically, will also be provided.

The Primary Subprograms

ASSGN

Function: The purpose of this subprogram is to process input and output data and to calculate necessary program constants.

Arguments: All arguments are transmitted by COMMON except

OMEGA = rotational velocity (Ω), RPM

VEL = wind speed (V), mps

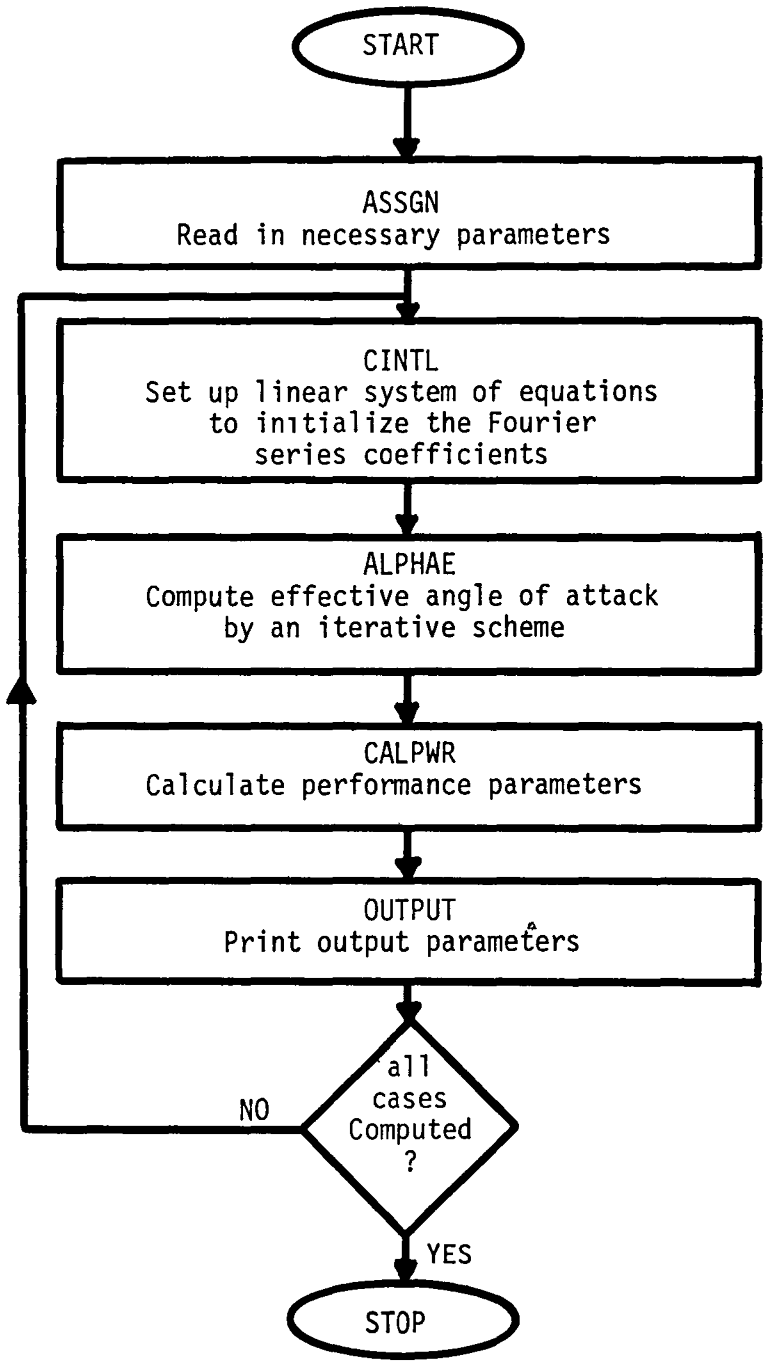

FIG. 3 FLOW DIAGRAM FROM THE COMPUTER PROGRAM VORTEX

SI = coning angle (ψ), degree

RDC = radians to degrees, conversion constant

NB = number of blades

PI = π

R = parameter defined by $R\cos\psi$

SKP1 = interval ahead of the singularity where the induction factor is interpolated

SKP2 = interval after the singularity where the induction factor is interpolated

Local Variables:

BT = twist angle (β), degree

Secondary subprograms called:

SKIP

Notes: To avoid recalculation of constants, all required constants have been calculated in this subroutine and transmitted to the main program. Also this subroutine calculates the parameters SKP1 and SKP2 if ISKP is set equal to 1; otherwise, the interval for interpolation around the singularity is read into the program.

CINTL

Function: The purpose of this subprogram is to establish the matrix equation to be solved for initializing the Fourier sine series coefficients. The coefficient matrix is constructed based on a linear lift line approximation.

Arguments: All arguments are transmitted by COMMON except

NP = number of points used in the spline interpolation and integration. This number must be at least 4 * N so that there will always be at least 4 points to obtain spline constants (subroutine SPLCOE will fail with less than 4 points).

ZETA = non-dimensional radial position along the blade

RDC = radians to degrees, a conversion constant

ALG = geometric angle of attack, α_g

TIGRL = matrix containing Fourier series terms

CA = coefficient matrix

RAS = known vector

Local variables:

DEL1 = a small number, 10^{-6}

YI = array containing the induction factors

Secondary Subprograms called:

TINGRL

ALPHAE

Function: The purpose of this subprogram is the calculation of the effective angle of attack, α_e, induced angle of attack, α_i, and circulation distribution, Γ, along the blade using an iteration method.

Arguments: All arguments transmitted by COMMON except

CA = coefficient matrix in matrix equation

RAS = known vector in matrix equation

TIGRL = matrix containing Fourier series terms

R = parameter defined by $R\cos\psi$

ALG = geometric angle of attack

ALE = effective angle of attack

ALI = induced angle of attack

GAMM = circulation distribution function

Local variables:

VR = undisturbed resultant velocity

ARG = argument of sine and cosine functions

MAXITR = maximum number of iterations

NITER = index of iteration loop

STIGR = parameter containing the integral part of the induced velocity.

Secondary Subprograms called:

CD230 and GAUSS

Notes: This subprogram contains the following error message - "convergence was not obtained within the specified number of iterations".

CALPWR

Function: The purpose of the subprogram is the computation of performance parameters, axial force, torque, power etc.

Arguments: All arguments are transmitted by COMMON except

ALE = effective angle of attack

R = parameter defined by $R\cos\psi$

Vel = wind velocity

AFRCE = distribution of axial force on the blade

TRQ = distribution of torque along the blade

SUMF = total axial force

SUMQ = total torque

CF = axial force coefficient

CP = power coefficient

RPWR = rotor power

APWR = alternator power

XRA = inverse of tip speed ratio

Local variables:

PHI = inflow angle (angle between resultant velocity and the axis of wind turbine).

VR = undisturbed velocity

CY = parameter defined by $C_L \cos\phi + C_D \sin\phi$

CX = parameter defined by $C_L \sin\phi - C_D \cos\phi$

RHB = hub radius

Secondary Subprograms called:

CD230, CD44, SPLINT

OUTPUT

Function: The purpose of this subprogram is to write out the performance parameters.

Arguments: All arguments are transmitted by COMMON except

ALE = effective angle of attack

ALI = induced angle of attack

OMEGA = rotational velocity (Ω), RPM

R = parameter defined by $R\cos\psi$

VEL = wind speed

AFRCE = axial force distribution

TRQ = torque distribution

SUMF = total axial force

SUMQ = total torque

CF = axial force coefficient

CP = power coefficient

RPWR = rotor power

APWR = alternate power

XRA = inverse of tip speed ratio

Secondary Subprograms called:

None

The following describes the supporting secondary subroutines of the main program.

AUX

Function: The purpose of this subroutine is the calculation of the integrand of the semi-infinite integral.

Arguments: All arguments are transmitted by COMMON except

THET = independent variable of the cylindrical coordinate system

FX = integrand of the integral

Secondary subroutines called:

None

CALCI

Function: The purpose of this subroutine is the calculation of the induction factor.

Arguments: All arguments are transmitted by COMMON except

XI = induction factor

NOPT = control parameter

(when NOPT = 2, calculation is performed for second blade

NOPT = 1, calculation is performed for first rotor.)

Secondary Subroutines called:

GLQUD

CD230

Function: The purpose of this subroutine is the calculation of the lift and drag coefficients using curve-fitted empirical data for NACA 230XX airfoils.

Arguments: All arguments are transmitted by COMMON except

IREN = control parameter

(when IREN = 0, no Reynolds number effect is considered,
when IREN = 1, airfoil characteristics are corrected for the Reynolds number)

ALPHA = effective angle of attack

CL = lift coefficient

CD = drag coefficient

X = non-dimensional radius

W = relative velocity

Local variables:

All airfoil parameters are referred to a smooth airfoil of 18% thickness to chord ratio at a Reynolds number of 3×10^6. Some of the parameters are shown schematically in Fig. 4.

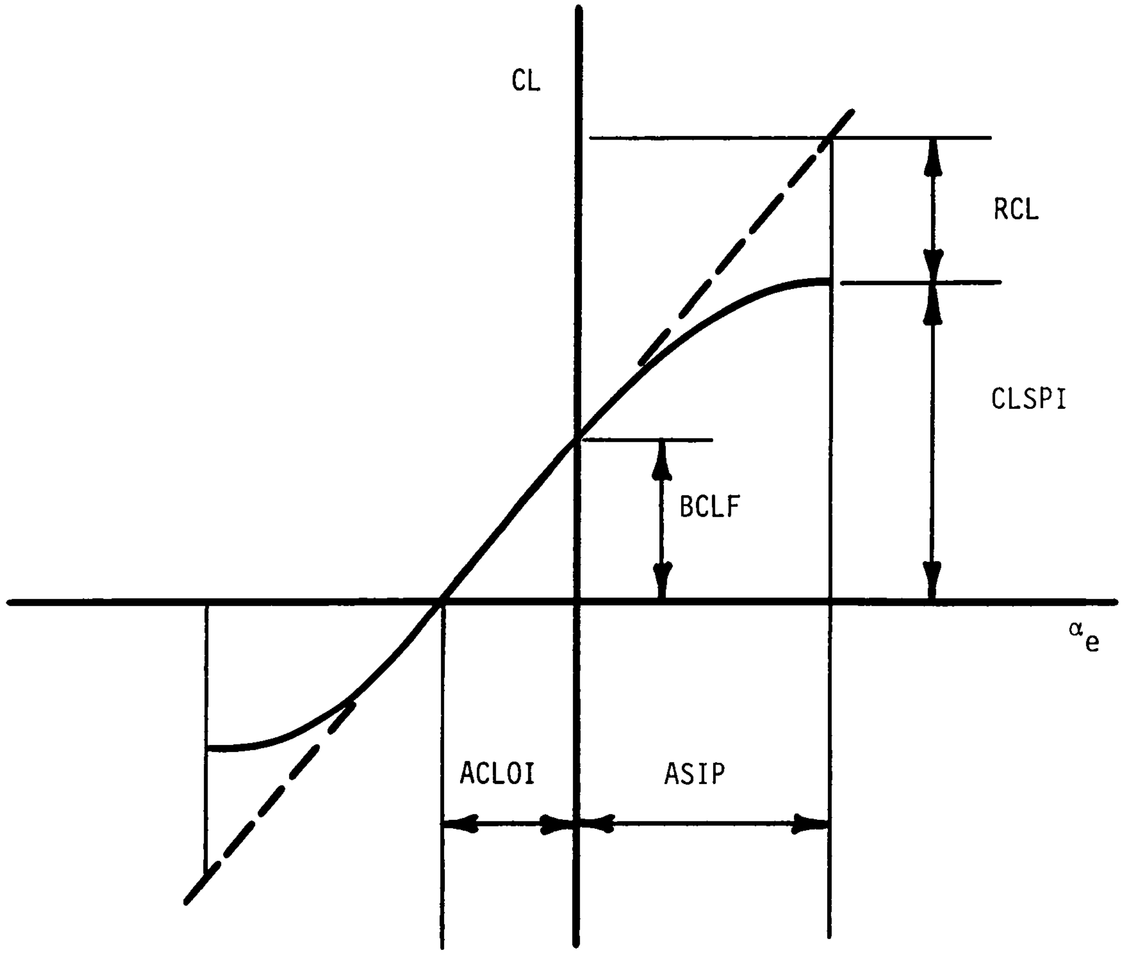

FIG. 4a LIFT COEFFICIENT CURVEFIT PARAMETERS

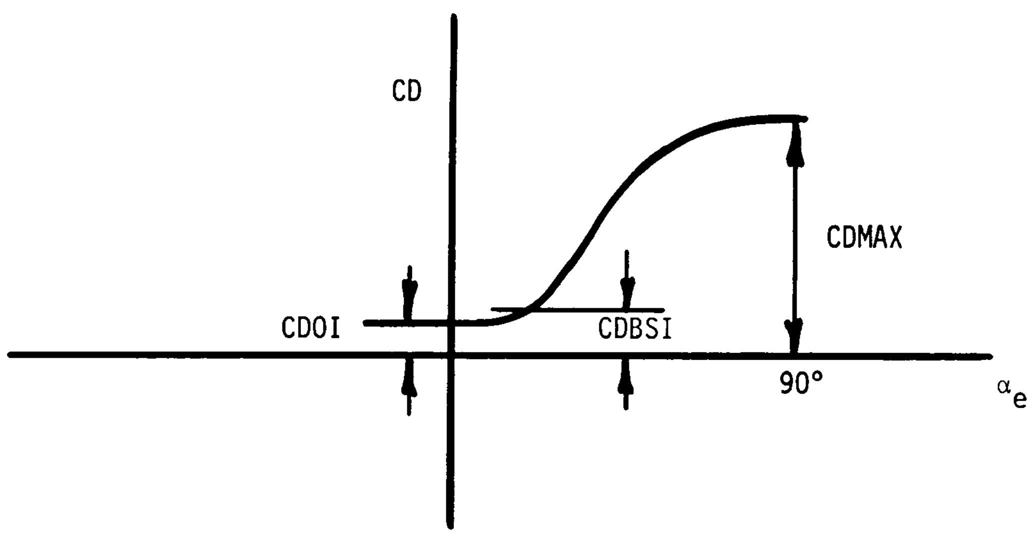

FIG. 4b DRAG COEFFICIENT CURVEFIT PARAMETERS

ACLOI = zero lift coefficient angle of attack

ASIP = stall angle of attack

CDBSI = drag coefficient at stall angle of attack smooth, 18% thickness to chord ratio and Reynolds number of 3×10^6.

CDOI = minimum drag coefficient

RCL = reduction in lift coefficient at stall angle

LEXP = power of the curvefit data after stall

SLI = slope of lift coefficient

ASCLOI = a constant, 90°

RENS = Reynolds number

FTL = slope correction factor for thickness to chord ratio and the Reynolds number effect.

FTD = drag coefficient correction for thickness to chord ratio and Reynolds number.

BCLF = zero incident angle of attack

ASFP = stall angle of attack corrected for thickness to chord ratio and Reynolds number.

ASFN = same as ASFP, but negative.

CLSPF = maximum lift coefficient

CLSNF = lift coefficient at ASFN

CDMAX = maximum drag coefficient

Secondary subroutines called:

None

Notes:

Airfoil characteristics are based on a two-dimensional, smooth, 18% thickness to chord ratio airfoil at a Reynolds number of 3×10^6. This data is corrected for sectional thickness to chord ratios and Reynolds numbers.

CD44

This subroutine is the same as CD230 except that it computes airfoil characteristics of the NACA 44XX airfoil series.

CTRWT

Function: The purpose of this subroutine is computation of the negative torque of a counter-weight for a single bladed wind turbine.

Arguments: All arguments are transmitted by COMMON except

V = wind speed

RHB = hub radius

CTRQ = negative torque of counter-weight

Secondary subroutines called:

SPLINT

FACTORS

Function: The purpose of this subroutine is the calculation of all induction factors.

Arguments: All arguments are transmitted by COMMON except

ZETA = Radial coordinates

NP = number of points used in the spline fit

J = index referring to the corresponding point on the blade

XHB = dimensionless hub radius

YI = induction factor

YIBZ = induction factor at ZETP-DEL

YIAZ = induction factor at ZETP+DEL

SKP1)
 } = interval around the singularity
 SKP2)

Secondary subroutines called:

CALCI, SPLCOE

GAUSS

Function: The purpose of this subroutine is the solution of a linear system of equations using the Gauss elimination technique.

Arguments: All arguments are transmitted by COMMON except

CA = coefficient matrix

M1 = number of equations

RQ = known vector

Local variables:

EPS = tolerance of the Gauss elimination technique

Secondary Subroutines called:

None

Notes:

This subroutine contains the following error message "error due to incorrect input of the number of equations".

GLQUD

Function: The purpose of this subroutine is for setting up the Gauss-Laguerre integration.

Arguments: All arguments are transmitted by COMMON except

AUX = auxiliary subroutine containing the integrand

ANS = computed integral

Local Variables:

X = Gauss-Laguerre roots

W = weighting factor

Secondary subroutines called:

AUX

SKIP

Function: The purpose of this subroutine is the calculation of the interval around the singularity where the induction factors are interpolated.

Arguments: All arguments are transmitted by COMMON except

ZETP = dimensionless radial position

VEL = wind velocity

OMEGA = rotational speed

RB = blade radius

SKP1)
     } = interval around the singularity
SKP2)

Secondary subroutines called:

None

Notes:

The intervals around the singularities are a part of input data to the program. This subroutine is provided to facilitate the evaluation of proper intervals. The procedure, however, is more based on the numerical experimentation than theoretical analysis. Unrealistic results may occur if improper intervals are used, therefore, care must be taken in application of this subroutine for tip speed ratios outside the range of cases considered herein. For such cases, the proper intervals can be found from the fact that the induction factor, by definition, is a smooth and continuous function of radial position.

SPLINT

Function: The purpose of this subroutine is the computation of definite integrals.

Arguments: All arguments are transmitted by COMMON except

ND = number of data points

ZZ = spanwise location

C = array containing the integrand

SUM = computed integral

Secondary subroutines called:

SPLCOE

SPLCOE

Function: The purpose of this subroutine is the calculation of the coefficients for the spline fit.

Arguments: All arguments are transmitted by COMMON except

XP = vector containing independent variable

YP = vector containing dependent variable

M = number of data points

Secondary subroutines called:

None

TINGRL

Function: The purpose of this subroutine is the calculation of the integral part of the induced velocity.

Arguments: All arguments are transmitted by COMMON except

NS = index referring to the corresponding spanwise station

NP = number of points on the blade for computing the integral

ZETA = non-dimensional radial coordinate

XHB = non-dimensional hub radius

DEL1 = a small number, 10^{-6}

YI = induction factor

YIBZ = induction factor at ZETP-DEL

YIAZ = induction factor at ZETP+DEL

TINT = integral part of the induced velocity

Secondary subroutines called:

SPLINT

COMMON AND I/O PARAMETERS

In this section, several important program details not covered in a description of all subprograms and a listing of program variables will be given. In particular, there will be a listing of terms that appear in COMMON followed by description of the input and output parameters of the program.

COMMON

Table 1 is a visual display of the subroutines in which COMMON appears.

COMMON	Subroutines where the common is used
AAA	AUX, CALI
BBB	MAIN, ASSGN, GLQUD
CCC	MAIN, ASSGN, CINTL, AUX, ALPHAE, TINGRL, CALPWR, CD230, CD44, FACTRS
DDD	MAIN, CD230, CD44, CTRW, CALPWR
EEE	MAIN, ASSGN, CINTL, ALPHAE, CD230, CD44
FFF	MAIN, ASSGN, CINTL, ALPHAE, CALPWR
GGG	MAIN, ASSGN, CALPWR
HHH	MAIN, ASSGN
OOO	ASSGN, CTRWT

TABLE 1

COMMON AAA:

contains the following parameters:

KK = index of loop on the number of blades

COMMON BBB:

contains the following parameters:

X = roots of Laguerre polynomial

W = weight factors for the Gauss-Laguerre quadrature

COMMON CCC:

contains the following parameters:

TT = radial location on the blade

TIP = radial location on the blade

PI = constant π

EL = tip speed ratio

NB = number of blades

COMMON DDD:

contains the following parameters:

RDC = radian to degrees conversion

WO = rotational speed

RHO = density

COMMON EEE:

contains parameters that are described in the input data section which follows. They remain unchanged during the execution of the program.

COMMON FFF:

Most parameters in this common are described in the following section on input parameters. The remaining parameters are:

ZETP = dimensionless radial coordinate along the blade (equally spaced from hub to tip)

BETA = twist angle distribution

COMMON GGG:

This common contains constants, calculated in the ASSGN subroutine, which are used throughout the main program and subroutines. These parameters are:

CSSI = cosine of the coning angle

COEF = a constant defined by $1/2\ \rho RN$

COEF1 = a constant defined by $\pi/2\ \rho R^2$

COMMON HHH:

This common contains parameters that are described in the input data section.

COMMON OOO:

This common contains information about the geometry of counter-weight of a single bladed rotor such as dimensions, coning angle, etc. All parameters are described in the section on input parameters.

DESCRIPTION OF THE INPUT PARAMETERS

IOP = output level control parameter

0: output does not include the input data

1: input data is first printed out then computed parameters are.

ISKP = control parameter

0: interpolation interval is input

1: interpolation interval is calculated

ITEM = control parameter

1: wind speed variable

2: rotational speed variable

IREN = control parameter

0: Reynolds' number effect not considered

1: Reynolds' number effect considered

MAXITR = maximum number of iterations

NCASES = number of problem cases considered in the computer run

N = number of stations along the blade. A maximum of 9 points is allowed (due to the machine storage limitation)

NPROF = parameter defining the type of airfoil being used. The present program is equipped with routines that can handle only two types of airfoils, NACA 230XX and NACA 44XX series. The following options are allowed:

NPROF = 23000 (230XX series)

NPROF = 4400 (44XX series)

AO = slope of lift coefficient per degree. This parameter should be input as accurately as possible because of the dependence of the initial guess of the iteration loop upon it.

BO = zero incident lift coefficient

DELT = increment of the loop on NCASES

NB = number of blades

RB = radius of the blades

OMEGA = rotational velocity, RPM. Initial rotational velocity if ITEM = 2.

VEL = wind speed, mps. Initial wind speed if ITEM = 1.

SI = coning angle, degree

TCR75 = thickness to chord ratio at 3/4 of span.

SLTCR = slope of the thickness to chord ratio (TCR) distribution. This implies that approximately linear distribution of TCR can be handled. However, non-linear distributions can be easily handled by replacing the linear equation for TCR calculation with the non-linear one.

CI75 = chord at 3/4 of the span

XX = array containing the spanwise stations where information about chord and twist angle is provided. This input data must be provided in dimensionless radius. The first station must be located where blade begins i.e., the dimensionless hub.

CHRD = array containing chord distribution. This data is to be provided at each XX location.

BT = twist angle distribution. This data is to be provided at each XX location.

X = array containing roots of Laguerre polynomial.

W = array containing weighting factor for Gauss-Laguerre integration.

Conditional Input Data

The following parameters are input when ISKP = 0

SKP1, SKP2 = interval around the singularity where the induction factors are interpolated.

The following parameters are input when NB = 1

(information about the size and location of different components of the counter-weight assuming an ellipsoidal counter-weight and cylindrical support span).

RA = radius at which counterweight is located

B1 = largest dimension of support spar

B2 = smallest dimension of support spar

B3 = dimension of counterweight (normal)

B4 = dimension of counterweight (in radial direction)

SIP = coning angle of spar

N2 = number of points on support spar for integration

N3 = number of points on counter-weight for integration

Description of the output.

The output is composed of two parts:

1) information pertaining to blade and operating condition.

This includes

- number of blades, blade radius, rpm
- pitch and chord distribution
- integration constants

2) distribution of computed parameters along the blade such as

- circulation
- induced angle of attack
- effective angle of attack
- axial force
- torque

followed by the performance parameters

- rotor power
- alternator power
- power coefficient
- axial force coefficient

A SAMPLE PROBLEM

The purpose of this section is to illustrate the use of VORTEX by way of an example problem. No attempt was made to select a problem that was trivial. Rather, the problem chosen was used because it is typical of most work done thus far. All input parameters needed to run VORTEX are displayed. The program output is presented as it comes from the machine. This output can serve as a means of deciding if the program is running correctly.

INPUT

The input data supplied the program is shown in Table 2. Explanation of this table is as follows:

The first row of the table is the input data called from the first READ statement in the subroutine ASSGN. The parameters read are in order:

IOP	= 1	(the input data will be printed before the computed parameters
ISKP	= 1	(the interpolation interval will be determined within the subroutine SKIP)
ITEM	= 1	(the calculations will be performed for different wind speeds for a fixed rotational speed)
IREN	= 0	(no Reynolds number correction of the airfoil data will be made
MAXITR	= 28	(a maximum of 28 iterations are allowed to converge the circulation distribution)
NCASES	= 1	(only one problem is being run)

N = 9 (there are 9 stations along the blade)

NPROF = 23 (the blade is made of a NACA 230XX airfoil)

DELT = 1.0 (if more than one problem were considered, the wind velocity [since ITEM = 1] would be increased by 1.0 mps)

The second row of Table 2 is the input data called from the second READ statement in the subroutine ASSGN. The parameters read are in order:

NB	= 2	(the machine considered has two blades)
RB	= 14.00	(the blade radius in meters)
OMEGA	= 40.0	(the rotor RPM)
VEL	= 12.0	(the wind velocity in meters per second)
SI	= 3.00	(the coning angle in degrees)
AO	= 0.0851	(the slope of the lift coefficient curve per degree)
BO	= 0.1030	(the lift coefficient at zero degrees of incidence)

The third row of Table 2 is the input data called from the third READ statement in the subroutine ASSGN. The parameters read are in order:

TCR75	= 0.240	(the thickness to chord ratio at 3/4 of the blade span
SLTCR	= 0.0920	(the slope of the distribution of the thickness to chord ratio)
CI75	= 1.520	(the chord length in meters at 3/4 of the blade span)

```
   1    1    1    0   28    1    9   23   1.0
   2   14.00    40.0     12.0     3.00       0.0851    0.1030
    .240     .0920    1.520
   0.31648D0       00.00D0         1.5200D0
   0.40008D0       00.00D0         1.5200D0
   0.50000D0       00.00D0         1.5200D0
   0.70000D0       00.00D0         1.5200D0
   0.80000D0       00.0000         1.5200D0
   0.82912D0       00.00D0         1.4230D0
   0.88608D0       00.00D0         1.2700D0
   0.94304D0       00.00D0         1.2200D0
   1.00000D0       00.00D0         1.1700D0
.81498279233948890D2          .55753457883283568D-34
.69962240035105030D2          .40883015936806578D-29
.61058531447218762D2          .24518188458784027D-25
.53608574544695070D2          .36057658645529590D-22
.47153106445156323D2          .20105174645555035D-19
.41451720484870767D2          .53501888130100376D-17
.36358405801651622D2          .78198003824594480D-15
.31776041352374723D2          .68941810529580857D-13
.27635971743327170D2          .39177365150584514D-11
.23887329848169733D2          .15070082262925849D-09
.20491460082616425D2          .40728589875499997D-08
.17417992646508979D2          .79608129591336300D-07
.14642732289596674D2          .11513158127372799D-05
.12146102711729766D2          .12544721977993333D-04
.99120980150777060D1          .10446121465927518D-03
.79275392471721520D1          .67216256409354789D-03
.61815351187367654D1          .33693490584783036D-02
.46650837034671708D1          .13226019405120157D-01
.33707742642089977D1          .40732478151408646D-01
.22925620586321903D1          .98166272629918890D-01
.14255975908036131D1          .18332268897777802D0
.76609690554593660D0          .25880670727286980D0
.31123914619848373D0          .25877410751742390D0
.59019852181507977D-1         .14281197333478185D0
```

TABLE 2

INPUT PARAMETERS

The fourth through twelfth rows in Table 2 are the input data called from the fourth READ statement in the subroutine ASSGN. The parameters read are in order:

XX(J), J = 1,...,9	(the first column; 9 spanwise locations along the blade: XX(9) = 1.000 is the blade tip)
BT(J), J = 1,...,9	(the second column; 9 values of blade twist angle)
CHRD(J), J = 1,...,9	(the third column; 9 values of the blade chord length in meters)

The last 24 rows in Table 2 are the input data called from the fifth READ statement in the subroutine ASSGN. The parameters read are in order:

X(L), L = 1,...,24	(the first 24 roots of the Laguerre polynomials)
W(L), L = 1,...,24	(the first 24 weighting factors to be used in the Gauss-Laguerre integration)

Output

Table 3 is a presentation of the output data of VORTEX. It can be seen that all input data has been written before the calculated parameters. After which the following arrays are presented in order:

nondimensional radial blade location	= ZETP(IH)
circulation	= GAMM(IH)
induced angle of attack	= ALI(IH)
effective angle of attack	= ALE(IH)
axial force on rotor	= AFRCE(IH)
rotor torque	= TRQ(IH)

The last row in Table 3 is a presentation of the computed output data.

VEL = 12.0	(wind speed, meters/second)
OMEGA = 40.0	(blade rotational speed, RPM)
XRA = 4.8802	(tip speed ratio: $\Omega R/V$)
SUMQ = 42812.612	(total torque on rotor, Newtons)
RPWR = 179.333	(rotor power, kw)
APWR = 170.366	(alternator power, kw)
CP = 0.276	(power coefficient)
SUMF = 26290.120	(total axial force on rotor, Newtons)
CF = 0.485	(axial force coefficient)

OPERATING CONDITIONS

NUMBER OF BLADES	2
RADIUS OF BLADE, m	14.00
CONING ANGLE, degree	3 0
ROTATIONAL SPEED, rpm	40 0
WIND SPEED, m/s	12.0
TYPE OF AIRFOIL	23

BLADE DATA

LIFT COEFF. SLOPE	0.085
ZERO INCIDENT LIFT	0.103
NUMBER OF STATIONS	9
THICKNESS @ 3/4 SPAN	0 24
THICKNESS DIST. SLOPE	0.09
CHORD @ 3/4 SPAN	1.52

LOCATION	TWIST(degree)	CHORD(m)
0.316480	0.000000	1.520000
0.400080	0 000000	1.520000
0.500000	0.000000	1 520000
0.700000	0.000000	1.520000
0.800000	0.000000	1 520000
0.829120	0.000000	1.423000
0 886080	0.000000	1.270000
0.943040	0.000000	1.220000
1.000000	0.000000	1.170000

INTEGRATION CONSTANTS

0.8149827923394889D+02	0.5575345788328357D-34
0.6996224003510503D+02	0.4088301593680658D-29
0.6105853144721876D+02	0.2451818845878403D-25
0.5360857454469507D+02	0.3605765864552959D-22
0.4715310644515632D+02	0.2010517464555503D-19
0.4145172048487077D+02	0.5350188813010038D-17
0.3635840580165162D+02	0.7819800382459448D-15
0.3177604135237472D+02	0.6894181052958086D-13
0.2763597174332717D+02	0.3917736515058451D-11
0.2388732984816973D+02	0.1507008226292585D-09
0.2049146008261642D+02	0.4072858987550000D-08
0.1741799264650898D+02	0.7960812959133630D-07
0.1464273228959667D+02	0 1151315812737280D-05
0.1214610271172977D+02	0.1254472197799333D-04
0.9912098015077706D+01	0 1044612146592752D-03
0.7927539247172152D+01	0.6721625640935479D-03
0.6181535118736765D+01	0 3369349058478304D-02
0.4665083703467171D+01	0.1322601940512016D-01
0 3370774264208998D+01	0.4073247815140865D-01
0.2292562058632190D+01	0.9816627262991889D-01
0.1425597590803613D+01	0.1833226889777780D+00
0.7660969055459366D+00	0.2588067072728698D+00
0.3112391461984837D+00	0.2587741075174239D+00
0.5901985218150798D-01	0.1428119733347818D+00

LOCATION	CIRCULATION	ALPHA I	ALPHA O	A.FORCE	TORQUE
0.31648E+00	0 00000E+00	-0.59498E+00	-0.21021E-01	-0.67172E+01	-0.49850E+03
0.40192E+00	0.21904E+02	-0.13449E+00	0 33643E+00	0.18568E+05	0 14388E+05
0.48736E+00	0 26060E+02	-0 18203E-01	0.37931E+00	0.27341E+05	0.24108E+05
0.57280E+00	0.33172E+02	-0.93133E-01	0.24998E+00	0 36660E+05	0.63859E+05
0.65824E+00	0 37414E+02	-0 52209E-01	0.24919E+00	0 48136E+05	0.98668E+05
0.74368E+00	0.38242E+02	-0.82225E-01	0.18629E+00	0.53644E+05	0.95743E+05
0.82912E+00	0.37602E+02	-0.67607E-01	0.17436E+00	0.60018E+05	0.11143E+06
0.91456E+00	0.28642E+02	-0.83125E-01	0.13699E+00	0 53593E+05	0.83884E+05
0.10000E+01	-0.13419E-14	-0.22293E+00	-0.21086E-01	-0 67986E+02	-0.79514E+04

MPS	RPM	XRATIO	TORQUE	POWER	ALT POWER	CP	A.FORCE	CF
12.0	40.0	4.8802	42812.612	179.333	170.366	0 276	26290.120	0 48

TABLE 3

OUTPUT PARAMETERS

CLOSURE

In this report, a computer program entitled VORTEX for the aerodynamic performance prediction of horizontal axis wind turbines has been presented. The program is fairly general as it can handle wind turbines with any number of blades having a variety of blade geometry and airfoil section. It has been found that the program VORTEX is relatively efficient: most problems are solved in less than a minute of IBM 4341 CPU time. Predicted results have been found in good agreement with existing experimental data.

REFERENCES

1. Jeng, D. R., Keith, Jr., T. G., and Aliakbarkhanafjeh, A., "Aerodynamic Analysis of a Horizontal Axis Wind Turbine by Use of Helical Vortex Theory, Volume I: Theory," NASA Contractor Report NASA CR-168054, December 1982.

2. Wilson, R. E. and Lissaman, P. S. B., "Applied Aerodynamics of Wind Power Machines," U.S. Department of Commerce, National Tech. Info. Service, PB 238-595, 1974.

3. Snyder, M. H. and Staples, D. L., "WIND II Users Manual," Wind Energy Lab, Wichita State University, WER-15, July 1982.

4. Jeng, D. R., Aliakbarkhanafjeh, A., Keith, Jr., T. G. and Peng, T., "Aerodynamic Performance and Load Analysis of the NASA Mod-0A and Mod-1 Wind Turbines," Proceedings of 1982 UMC Wind/Solar Engineering Conference, pp. 67-74. Kansas City, Missouri, April 1982.

5. White, J. A., "Performance Analysis of a Tip-Controlled Wind Turbine Utilizing a Helical Vortex Wake Model," MS Thesis, University of Toledo, May 1983.

6. White, J. A., Keith, Jr., T. G. and Jeng, D. R., "Performance Prediction of a Tip-Controlled Wind Turbine Rotor," Proceedings of the 1983 UMC Wind/Solar Engineering Conference, Kansas City, Missouri, April 1983.

7. Keith, Jr., T. G., White, J. A. and Jeng, D. R., "Aerodynamic Performance Prediction of a Tip-Controlled Horizontal Axis Wind Turbine," Proceedings of Wind Workship VI, ASES Annual Meeting, Minneapolis, Minnesota, May 1983.

8. Corrigan, R. D., Ensworth, C. B. and Keith, Jr., T. G., "Performance Comparison Between NACA 23024 and NACA 64_3618 Airfoil Configured Rotors for Horizontal Axis Wind Turbines," Proceedings of Wind Workshop VI, ASES Annual Meeting, Minneapolis, Minnesota, May 1983.

9. McCormick, B. W., Aerodynamics of V/STOL Flight, Academic Press, New York, New York, 1967.

10. Pennington, R. H., Introductory Computer Methods and Numerical Analysis, MacMillan Co., London, 1965.

11. Shao, C. F., "Tables of Zeros and Gaussian Weights of Certain Associated Polynomials and Related Generalized Hermite Polynomials," IBM Technical Report TR00.1100, March 1964.

12. Carnahan, B., Luther, H. A., and Wilkes, J. O., Applied Numerical Methods, John Wiley and Sons, Inc., New York, New York, 1969.

13. Moriya, T., "On the Induced Velocity and Characteristics of the Propellor," Selected Scientific and Technical Papers, University of Tokyo, 1959.

PROGRAM LISTING

```
      SUBROUTINE AAA
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     THIS PROGRAM CALCULATES THE THEORETICAL AERODYNAMIC PERFORMANCE
C     PARAMETERS OF A HORIZONTAL AXIS WIND TURBINE USING A HELICAL
C     VORTEX METHOD. THE CONTROL PARAMETERS ARE:
C
C     IOP=1         STANDARD OUTPUT
C     IOP=0         STANDARD AS WELL AS INPUT DATA
C     ITEM=1          WIND SPEED IS VARIABLE
C     ITEM=2          ROTATIONAL SPEED IS VARIABLE
C
C
C     THE IMPORTANT INPUT PARAMETERS ARE:
C
C     AO        LIFT CURVE SLOPE
C     BO        ZERO INCIDENT LIFT COEFFICIENT
C     DEL       THE INTERVAL AROUND SINGULARITY THAT IS INTEGRATED
C               SEPARATELY.
C     DELT      THE INCREMENTAL CHANGE IN PARAMETRIC RUN
C     EL        THE TIP SPEED RATIO, V/WO*R
C     EPS       A SMALL NUMBER, TOLERANCE OF GAUSS ELIMINATIO
C               METHOD
C     NCASES    NUMBER OF CASES IN PARAMETRIC RUN
C     N         NUMBER OF POINTS ON THE BLADE
C     NB        NUMBER OF BLADES
C     NP        NUMBER OF POINTS USED IN THE SPLINE INTERPOLATION
C               AND INTEGRATION. IT MUST BE AT LEAST 4*N SO THAT
C               WHEN INTEGRATING FROM X=0 TO ZETP-DEL OR FROM ZETP+DEL
C               TO 1, THERE WILL ALWAYS BE AT LEAST 4 POINTS FOR
C               THE SPLINE INTEGRATION.(SUBROUTINE SPLCOE WILL
C               FAIL WITH LESS THAN FOUR POINTS)
C     OMEGA     ROTATIONAL VELOCITY, RPM
C     PR        RATED POWER OF WIND TURBINE, KW
C     RB        ROTOR RADIUS, FT
C     RA        RADIUS AT WHICH COUNTER-WEIGHT IS LOCATED
C     SI        CONING ANGLE, DEGREES
C     SIP       CONING ANGLE OF SPAR SUPPORT, DEGREES
C     SKP1(J)   BLADEWISE LOCATIONS BETWEEN WHICH THE INDUCTION
C     SKP2(J)   FACTOR IS INTERPOLATED
C     VEL       WIND VELOCITY, MPH
C     X,W       LAGUERRE-GAUSSIAN INTEGRATION CONSTANTS.
C     ZETP      NON-DIMENSIONAL DISTANCE ALONG THE BLADE WHERE
C               CALCULATIONS HAVE BEEN PERFORMED FOR FOURIER
C               SERIES COEFFICIENTS.
C
C
C     .................... MAIN PROGRAM ..........................
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ALE(9), ALI(9), RAS(9),
     1           ALG(9 ),ZETA(81),
     2          CA(9 ,9), AS(9), TIGRL(9,9),
```

```
     3           GAMM(9),AFRCE(9),TRQ(9)
     5           ,SKP1(9),SKP2(9)
      COMMON/BBB/X(24), W(24)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      COMMON/DDD/ RDC,WO,RHO
      COMMON/EEE/ RB,TCR75,SLTCR,CI75
      COMMON/FFF/ N,NPROF,AO,BO,ZETP(9),BETA(9),C(9)
      COMMON/GGG/ CSSI,COEF,COEF1
      COMMON/HHH/ MAXITR,NCASES,ITEM,DELT
      DATA NCASES,DELT,ITEM/5,2.DO,1/
      DATA IN,IO/3,8/
      OPEN(NAME='RTPM.DAT',TYPE='OLD',READONLY,UNIT=IN)
C     RADIANS TO DEGREES CONVERSION
C     READ IN AND PRINT OUT NECESSARY PARAMETERS
C
      CALL ASSGN(OMEGA,VEL,PI,NB,R,SKP1,SKP2,RDC)
C
      NUMB=1
      NP=(N-1)*10+1
      DO 10 I=1,NP
   10 ZETA(I)=FLOAT(I-1)/FLOAT(NP-1)*(1.-ZETP(1))+ZETP(1)
   20 V=VEL
      WO=OMEGA*PI/30.DO
      EL=V/RB/WO
C
C     CALCULATE THE INITIAL VALUE FOR A'S
      CALL CINTL(SKP1,SKP2,NP,ZETA,RDC,ALG,R,WO,TIGRL
     1,CA,RAS)
C     THIS SECTION COMPUTES ALE VALUES USING AN ITERATION PROCESS
C
      CALL ALPHAE(CA,RAS,TIGRL,R,WO,MAXITR,ALG,ALI,ALE,GAMM)
C
C
      CALL CALPWR(ALE,R,RB,V,AFRCE,TRQ,SUMF,SUMQ,CF,CP,RPWR
     1,APWR,XRA)
C
C
C     PRINT OUT STANDARD OUTPUT
      CALL OUTPUT(N,ZETP,GAMM,ALE,ALI,AFRCE,TRQ,VEL,OMEGA,XRA,SUMQ
     1,RPWR,APWR,CP,SUMF,CF)
      NUMB=NUMB+1
      IF(NUMB.GT.NCASES) GO TO 30
      IF(ITEM.EQ.1) VEL=VEL+DELT
      IF(ITEM.EQ.2) OMEGA=OMEGA+DELT
      GO TO 20
   30 CONTINUE
      CLOSE( UNIT=IN )
      STOP
      END
```

```
C
C     ................ SUBROUTINES OF THE PROGRAM ................
C
      SUBROUTINE ASSGN(OMEGA,VEL,PI,NB,R,SKP1,SKP2,RDC)
C     THIS SUBROUTINE READS IN AND PRINTS OUT THE INPUT DATA
C
C     THE OUTPUT PARAMETERS IN THE ARGUMENT ARE:
C     OMEGA     ROTATIONAL VELOCITY, RPM
C     VEL       WIND SPEED
C     SI        CONING ANGLE, DEGREES
C     RDC       RADIANS TO DEGREES CONVERSION
C     MAXITR    MAXIMUM NUMBER OF ITERATIONS
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SKP1(9),SKP2(9),CHRD(9),BET(9),XX(9)
      COMMON/BBB/X(24), W(24)
      COMMON/EEE/ RB,TCR75,SLTCR,CI75
      COMMON/FFF/ N,NPROF,AO,BO,ZETP(9),BETA(9),C(9)
      COMMON/GGG/ CSSI,COEF,COEF1
      COMMON/HHH/ MAXITR,NCASES,ITEM,DELT
      COMMON/OOO/ RA,B1,B2,B3,B4,SIP,N2,N3
      DATA IN,IO/3,8/
      RDC=57.2957795130823DO
      RHO=1.2254710DO
      PI=3.14159265358979DO
      READ(IN,100) IOP,ISKP,ITEM,IREN,MAXITR,NCASES,N,NPROF,DELT
      READ(IN,110) NB,RB,OMEGA,VEL,SI,AO,BO
      IF(IOP.EQ.1) WRITE (IO,120) NB,RB,SI,OMEGA,VEL,NPROF
      READ (IN,130) TCR75,SLTCR,CI75
      IF(IOP.EQ.1) WRITE(IO,150) AO,BO,N,TCR75,SLTCR,CI75
C
C     READ IN BLADE ANGLE, CHORD, AND CORRESPONDING LOCATIONS
C     ON THE BLADE
C
      DO 10 J=1,N
      READ(IN,160) XX(J), BT, CHRD(J)
      IF(IOP.EQ.1) WRITE(IO,170) XX(J),BT,CHRD(J)
      BET(J)=BT/RDC
   10 CONTINUE
C     READ IN DATA FOR A 24 POINT LANGUERRE-GAUSS INTEGRATION
      READ(IN,140)  (X(L), W(L), L=1,24)
      IF(IOP.EQ.1) WRITE(IO,220)
      IF(IOP.EQ.1) WRITE(IO,230) (X(I),W(I),I=1,24)
      ZETP(1)=XX(1)
      C(1)=CHRD(1)
      BETA(1)=BET(1)
      J=2
      DO 20 I=2,N
      ZETP(I)=(1.DO-ZETP(1))/FLOAT((N-1))*(I-1)+ZETP(1)
   20 CONTINUE
      DO 40 I=2,N
      IF(ZETP(J).LE.XX(I).AND.ZETP(J).GE.XX(I-1)) GO TO 30
      GO TO 40
   30 DL=XX(I)-XX(I-1)
      DLP=ZETP(J)-XX(I)
```

```
      C(J)=(CHRD(I)-CHRD(I-1))/DL*DLP+CHRD(I)
      BETA(J)=(BET(I)-BET(I-1))/DL*DLP+BET(I)
      J=J+1
      IF(ZETP(J).LE.XX(I).AND.ZETP(J).GE.XX(I-1)) GO TO 30
   40 CONTINUE
   62 DO 61 J=1,N
C       IF(IOP.EQ.1) WRITE(IO,170) ZETP(J),BETA(J),C(J)
   61 CONTINUE
      IF(ISKP.EQ.1) GO TO 50
      READ(IN,12) (SKP1(J), SKP2(J),J=1,N)
      GO TO 60
   50 CALL SKIP(ZETP,RB,OMEGA,VEL,N,SKP1,SKP2)
   60 CSSI=DCOS(SI/RDC)
      R=RB*CSSI
      COEF=0.5*NB*RB*RHO
      COEF1=0.5*RHO*PI*R**2
      AO=AO*RDC
      IF(NB.NE.1) GO TO 70
C     READ IN COUNTER-WEIGHT DIMENSIONS
C     SIP IS THE CONING ANGLE OF THE SPAR SUPPORT
      READ(IN,190) RA,B1,B2,B3,B4,SIP
C     N2 A PARAMETER WHERE N2-1 IS THE NUMBER OF SUBDOMAINS IN SPAR
C     SUPPORT FOR SPLINE INTEGRATION
C     N3 SAME AS N2 EXCEPT THAT N3-1 IS FOR COUNTER-WEIGHT
      READ(IN,180) N2,N3
      IF(IOP.EQ.1) WRITE(IO,200) N2,RA,SIP,B1,B2
      IF(IOP.EQ.1) WRITE(IO,210) N3,B3,B4
   70 CONTINUE
C
C     ................. INPUT AND OUTPUT FORMATS ................
C
  100 FORMAT (8I5,F10.3)
  110 FORMAT(I5,6F10.5)
  120 FORMAT(///41X,'OPERATING CONDITIONS',/41X,'--------------------'
     1,/35X,'NUMBER OF BLADES',10X,I5,/35X,'RADIUS OF BLADE, m'
     2,6X,F7.2,/35X,'CONING ANGLE, degree',6X,F5.1,/35X,'ROTATIONAL'
     3,' SPEED, rpm',5X,F5.1,/35X,'WIND SPEED, m/s',11X,F5.1,/35X
     4,'TYPE OF AIRFOIL',11X,I5///)
  130 FORMAT (3F10.4)
  140 FORMAT (2D30.17)
  150 FORMAT(45X,'BLADE DATA',/45X,'----------',/35X,'LIFT '
     1,'COEFF. SLOPE',9X,F5.3,/35X,'ZERO INCIDENT LIFT',8X,F5.3
     2,/35X,'NUMBER OF STATIONS',8X,I5,/35X,'THICKNESS'
     3,' @ 3/4 SPAN',6X,F5.2,/35X,'THICKNESS DIST. SLOPE'
     4,5X,F5.2,/35X,'CHORD @ 3/4 SPAN',10X,F5.2,//29X,'LOCATION'
     5,6X,'TWIST(degree)',5X,'CHORD(m)')
  160 FORMAT(3F16.6)
  170 FORMAT(21X,3F16.6)
   12 FORMAT(2F10.5)
  180 FORMAT(2I5)
  190 FORMAT(6F8.3)
  200 FORMAT(/45X,'SPAR DATA',/45X,'---------'/35X,'NUMBER OF STATIONS'
     1,7X,I5,/35X,'RADIUS, m',25X,F5.2,/35X,'B1 DIMENSION, m',10X,F5.2,
     2/35X,'B2 DIMENSION, m',10X,F5.2)
```

```
210 FORMAT(/41X,'COUNTER-WEIGHT DATA',/41X,'-------------------'
   1,/35X,'NUMBER OF STATIONS'7X,I5,/35X,'B3 DIMENSION, m',10X,F5.2,
   2/35X,'B4 DIMENSION, m',10X,F5.2)
220 FORMAT(/42X,'INTEGRATION CONSTANTS',/42X,'---------------------')
230 FORMAT(8X,2D35.16)
    RETURN
    END
```

```
C                    '
C
C     ..........................................................
C
      SUBROUTINE CINTL(SKP1,SKP2,NP,ZETA,RDC,ALG,R,WO
     1,TIGRL,CA,RAS)
C     ..... THIS SUBROUTINE COMPUTE THE INITIAL VALUES OF ¢A| .....
C     THE INPUT PARAMETERS IN THE ARGUMENT ARE:
C     NP        NUMBER OF POINTS USED IN THE SPLINE INTERPOLATION
C     ZETA      RADIAL COORDINATES
C     SI        CONING ANGLE, DEGREES
C     ALG       GEOMETRIC ANGLE OF ATTACK
C     R         PARAMETER DEFINED BY RB*COS(SI)
C     THE OUTPUT PARAMETERS IN THE ARGUMENT ARE:
C     TIGRL     MATRIX CONTAINING THE INTEGRAL FROM HUB TO TIP
C     CA        COEFFICIENT MATRIX ¢A|
C     RAS       KNOWN VECTOR IN THE MATRIX EQUATION
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION IS(81),YI(81),TIGRL(9,9),CA(9,9)
     1,RAS(9),ALG(9),ZETA(81)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      COMMON/EEE/ RB,TCR75,SLTCR,CI75
      COMMON/FFF/ N,NPROF,AO,BO,ZETP(9),BETA(9),C(9)
      DEL1=0.000001D0
      XHB=ZETP(1)
      DO 10 J=1,N
C     ALG IS THE GEOMETRIC ANGLE OF ATTACK
      IF(ZETP(J).LE.0.001D0) ALG(J)=-BETA(J)+PI/2.0D0
      IF(ZETP(J).GT.0.001D0) ALG(J)=-BETA(J)+DATAN(EL/ZETP(J))
   10 CONTINUE
      DO 40 J=1,N
      DO 20 I=1,NP
      IS(I)=0
   20 YI(I)=0.0D0
      TTP=ZETP(J)
      SH=EL**2+ZETP(J)**2
C
      CALL FACTRS(SKP1,SKP2,IS,ZETA,NP,J,XHB,YI,YIBZ,YIAZ)
C
C      WRITE(7,2002) (YI(I),I=1,NP)
C 2002 FORMAT(5F10.4)
      DO 30 NS=1,N
C
C     ...... CALCULATE THE INTEGRAL FROM XHB TO 1 ......
C
      CALL TINGRL(NS,NP,ZETA,XHB,DEL1,YI,YIBZ,YIAZ,TINT)
      TIGRL(J,NS)=TINT
      ARG=PI*(ZETP(J)-XHB)/(1.0D0-XHB)
      T=-SLTCR*(ZETP(J)-0.75)+TCR75
      SLC=AO+(0.18-T)*0.0050*RDC
C     SET UP THE MATRIX EQUATION
      CA(J,NS)=2.DO*R*DSIN(NS*ARG)/AO/C(J)-NS*C(J)*TINT/4.DO/(1.DO-
     1XHB)
   30 CONTINUE
```

```
      T=-SLTCR*(ZETP(J)-0.75)+TCR75
      RAS(J)=WO*R**2*DSQRT(SH)*(BO/AO+ALG(J)+.005*(.18-T)/SLC)*C(J)
   40 CONTINUE
      RETURN
      END
```

```
C
C     .......................................................
C
      SUBROUTINE ALPHAE(CA,RAS,TIGRL,R,WO,MAXITR,ALG,ALI,ALE,GAMM)
C     ..... THIS SUBROUTINE CALCULATES EFFECTIVE ANGLE OF ATTACK BY
C           AN ITERATION SCHEME .....
C     THE INPUT PARAMETERS IN THE ARGUMENT ARE:
C     CA        COEFFICIENT MATRIX ¢A|
C     RAS       KNOWN VECTOR IN THE MATRIX EQUATION
C     TIGRL     MATRIX CONTAINING THE INTEGRAL FROM HUB TO TIP
C     R         PARAMETER DEFINED BY RB*COS(SI)
C     MAXITR    MAXIMUM NUMBER OF ITERATIONS
C     ALG       GEOMETRIC ANGLE OF ATTACK
C     THE OUTPUT PARAMETERS IN THE ARGUMENT ARE:
C     ALE       EFFECTIVE ANGLE OF ATTACK
C     ALI       INDUCED ANGLE OF ATTACK
C     GAMM      CIRCULATION
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AS(9),GAMM(9),IS(81),TIGRL(9,9),CA(9,9),RAS(9)
     1,ALG(9),ALE(9),ALI(9)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      COMMON/EEE/ RB,TCR75,SLTCR,CI75
      COMMON/FFF/ N,NPROF,AO,BO,ZETP(9),BETA(9),C(9)
      DATA IO/8/
      RDC=57.2957795130823DO
      NITER=0
      XHB=ZETP(1)
      GO TO 40
   10 NITER=NITER+1
      DO 30 J=1,N
      VR=WO*R*DSQRT(EL**2+ZETP(J)**2)
      ARG=PI*(ZETP(J)-XHB)/(1.0DO-XHB)
C     REVISE MATRIX EQUATION FOR ITERATION PROCESS
      DO 20 NS=1,N
      IF(J.EQ.1 .OR. J.EQ.N) CA(J,NS)=-TIGRL(J,NS)*NS/4.DO/(1.DO-XHB)
      IF(J.EQ.1 .OR. J.EQ.N) GO TO 20
      CA(J,NS)=DSIN(NS*ARG)
   20 CONTINUE
      IF(J.EQ.1 .OR. J.EQ.N) GO TO 30
      RAS(J)=GAMM(J)
   30 CONTINUE
   40 DO 50 J=1,N
   50 AS(J)=RAS(J)
C     SOLVE MATRIX EQUATION USING GAUSS ELIMINATION TECHNIQUE
      CALL GAUSS  (CA, N, AS)
      IF(NITER.NE.0) GO TO 80
      DO 70 NK=1,N
      VR=WO*R*DSQRT(EL**2+ZETP(NK)**2)
      ARG=PI*(ZETP(NK)-XHB)/(1.0DO-XHB)
      GAMM(NK)=0.0DO
      DO 60 K=1,N
C     CALCULATE THE CIRCULATION
   60 GAMM(NK)=GAMM(NK)+AS(K)*DSIN(K*ARG)
      T=-SLTCR*(ZETP(NK)-0.75)+TCR75
```

```
      SLC=AO+(0.18-T)*0.0050*RDC
C     CALCULATE THE EFFECTIVE ANGLE OF ATTACK
      ALE(NK)=2.0D0*GAMM(NK)/(AO*C(NK)*VR)-BO/SLC-.005*(.18-T)/SLC
      LST=N+NK
      IF(ALE(NK) .LE.0.04100D0) IS(LST)=2
      IF(NK.EQ.1.OR.NK.EQ.N) GO TO 70
      IF(ALE(NK).GE.0.25) ALE(NK)=ALG(NK)-.01
   70 CONTINUE
   80 DO 140 NK=1,N
      LST=N+NK
      IS(NK)=0
      STIGR=0.0D0
      VR=WO*R*DSQRT(EL**2+ZETP(NK)**2)
      DO 90 K=1,N
   90 STIGR=STIGR+TIGRL(NK,K)*AS(K)*K
      IF(NITER.GE.1 .AND.(NK.EQ.1.OR.NK.EQ.N)) GO TO 100
      ALI(NK)=STIGR/(4.D0*R*VR*(1.D0-XHB))
      ALE(NK)=ALG(NK)+ALI(NK)
  100 AL=ALE(NK)
      XB=ZETP(NK)
      IF(NPROF .EQ. 44) GO TO 110
      CALL CD230(IREN,AL,XB,VR,CL,CD)
      GO TO 120
  110 CALL CD44(AL,CD,CL,XB,VR)
  120 CLO=2.D0*GAMM(NK)/VR/C(NK)
      DELCL=(CL-CLO)/(CL+.00001)
      IF(DABS(DELCL).GT..02)GO TO 130
      GO TO 140
  130 IF(IS(LST).EQ.2) GO TO 140
      IF(NK.EQ.1.OR.NK.EQ.N) GO TO 140
      GAMM(NK)=.25D0*C(NK)*(CL+CLO)*VR
      IS(NK)=1
  140 CONTINUE
      SIA=0.D0
      DO 150 I=1,N
  150 SIA=SIA+IS(I)
      IF(SIA.LT.1) GO TO 170
      IF(NITER.GT.MAXITR) GO TO 160
      GO TO 10
C     ITERATIOIN PROCESS IS NOW COMPLETED
  160 WRITE(IO,200)
      STOP
  170 RETURN
  200 FORMAT('=== NO. OF ITERATION EXEEDED THE LIMIT ===')
      END
```

```
C
C     ........................................................
C
      SUBROUTINE CALPWR(ALE,R,RB,V,AFRCE,TRQ,SUMF,SUMQ,CF,CP
     1,RPWR,APWR,XRA)
C     THIS SUBROUTINE COMPUTES FORCE, TORQUE AND POWER
C
C     THE INPUT PARAMETERS IN THE ARGUMENT ARE:
C     ALE       EFFECTIVE ANGLE OF ATTACK
C     R         PARAMETER DEFINED BY RB*COS(SI)
C     SI        CONING ANGLE, DEGREES
C     RB        RADIUS OF THE BLADE
C     V         WIND SPEED
C     THE OUTPUT PARAMETERS IN THE ARGUMENT ARE:
C     AFRCE     LOCAL AXIAL FORCE ON THE BLADE
C     TRQ       LOCAL TORQUE ON THE BLADE
C     SUMF      TOTAL AXIAL FORCE
C     SUMQ      TOTAL TORQUE
C     CF        AXIAL FORCE COEFFICIENT
C     CP        COEFFICIENT OF PERFORMANCE
C     RPWR      TOTAL POWER
C     APWR      ALTERNATOR POWER
C     XRA       INVERSE OF TIP SPEED RATIO
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ALE(9),AFRCE(9),TRQ(9)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      COMMON/DDD/ RDC,WO,RHO
      COMMON/FFF/ N,NPROF,AO,BO,ZETP(9),BETA(9),C(9)
      COMMON/GGG/ CSSI,COEF,COEF1
 1600 DO 1750 NK=1,N
      PHI=BETA(NK)+ALE(NK)
      XB=ZETP(NK)
      VR=WO*R*DSQRT(EL**2+ZETP(NK)**2)
      ALOO=ALE(NK)
      IF(NPROF .EQ. 44) GO TO 1650
      CALL CD230(IREN,ALOO,XB,VR,CL,CD)
      GO TO 1700
 1650 CALL CD44(ALOO,CD,CL,XB,VR)
 1700 CY=CL*DCOS(PHI)+CD*DSIN(PHI)
      CX=CL*DSIN(PHI)-CD*DCOS(PHI)
      COQT=COEF*C(NK)*VR*VR
C     THIS SECTION CALCULATES AXIAL FORCE AND TORQUE AT N LOCATIONS
C     ALONG THE BLADE
      TRQ(NK)=COQT*R*ZETP(NK)*CX
 1750 AFRCE(NK)=COQT*CY*CSSI
C     INTEGRATE THE AXIAL FORCE USING SPLINE INTERPOLATION FORMULA
      SUMF=0.0D0
      CALL SPLINT(N,ZETP,AFRCE,SUMF)
C     INTEGRATE THE TORQUE OVER THE BLADE USING SPLINE INTERPOLATION
C     INTERPOLATION FORMULA
      SUMQ=0.0D0
      CALL SPLINT(N,ZETP,TRQ,SUMQ)
C
C     CALCULATE THE EFFECT OF COUNTER-WEIGHT
```

```
C
      IF(NB.NE.1) GO TO 1850
      RHB=RB*ZETP(1)
      CALL CTRWT(V,RHB,CTRQ)
 1850 CF=SUMF/(COEF1*V**2)
      SUMQ=SUMQ+CTRQ
      RPWR=SUMQ*WO/1000.DO
C     COMPUTE ALTERNATOR POWER
      APWR=.95DO*(RPWR-.075*PR)
      CP=SUMQ*WO/(COEF1*V**3)
      XRA=WO*R/V
      RETURN
      END
```

```
C
C     ............................................................
C
      SUBROUTINE OUTPUT(N,ZETP,GAMM,ALE,ALI,AFRCE,TRQ,VEL,OMEGA,XRA,SUMQ
     1,RPWR,APWR,CP,SUMF,CF)
C     ..... THIS SUBROUTINE PRINTS OUT THE STANDARD OUTPUT .....
C     THE INPUT PARAMETERS IN THE ARGUMENT ARE:
C     N         NUMBER OF TERMS IN THE SINE SERIES
C     ZETP      DIMENSIONLESS DISTANCE ALONG THE BLADE WHERE CALCULATION
C               HAVE BEEN PERFORMED FOR FOURIER SERIES COEFFICIENTS
C     GAMM      CIRCULATION
C     ALE       EFFECTIVE ANGLE OF ATTACK
C     ALI       INDUCED ANGLE OF ATTACK
C     AFRCE     LOCAL AXIAL FORCE ON THE BLADE
C     TRQ       LOCAL TORQUE ON THE BLADE
C     VEL       WIND SPEED
C     OMEGA     ROTATIONAL VELOCITY, RPM
C     XRA       INVERSE OF TIP SPEED RATIO
C     SUMQ      TOTAL TORQUE
C     RPWR      TOTAL POWER
C     APWR      ALTERNATOR POWER
C     CP        COEFFICIENT OF PERFORMANCE
C     SUMF      TOTAL AXIAL FORCE
C     CF        AXIAL FORCE COEFFICIENT
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ZETP(9),GAMM(9),ALE(9),ALI(9),AFRCE(9),TRQ(9)
      DATA IO/8/
      WRITE(IO,2120)
      DO 1800 IH=1,N
 1800 WRITE(IO,2130) ZETP(IH), GAMM(IH), ALI(IH), ALE(IH), AFRCE(IH),
     1  TRQ(IH)
      WRITE(IO,2140)
      WRITE(IO,2150) VEL,OMEGA,XRA,SUMQ,RPWR,APWR,CP,SUMF,CF
C
C    ................. STANDARD OUTPUT FORMATS ..............
C
 2120 FORMAT (//' ',7X, 'LOCATION',8X,'CIRCULATION',8X,'ALPHA I'
     1             , 10X, 'ALPHA O', 11X, 'A.FORCE', 11X, 'TORQUE')
 2130 FORMAT(' ',6E17.5)
 2140 FORMAT (/7X,'MPS',6X,'RPM',9X,'XRATIO',9X,'TORQUE',12X,
     1'POWER',9X,'ALT.POWER',5X,'CP',4X,'A.FORCE',7X,'CF')
 2150 FORMAT(2F10.1,F15.4,3F16.3,3F10.3,//)
      RETURN
      END
```

```
C
C     ...................................
C
      SUBROUTINE CALCI(XI,NOPT)
C     THIS SUBROUTINE CALCULATES INDUCTION FACTORS AS FOLLOWS:
C     NOPT = 1 CALCULATES I1+I2=I
C     NOPT = 2 CALCULATES THE INTEGRAL OF M FROM 0 TO INFINITY
C     WHERE I2=(ZETA-ZETP)*P2
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/AAA/ KK
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      DIMENSION AN(2)
      EXTERNAL AUX
      IF(NB.EQ.1) GO TO 50
      DO 10 KK=1,2
      IF(NOPT.EQ.2 .AND. KK.EQ.2) GO TO 20
      CALL GLQUD(AUX,ANS)
   10 AN(KK)=ANS
      IF(NOPT.EQ.1 .AND. DABS(TT-TTP).LE.0.001D0) XI=1.0D0
      IF(NOPT.EQ.1 .AND. DABS(TT-TTP).GT.0.001D0)
     1        XI=(AN(1)+AN(2))*(TT-TTP)
   20 IF(NOPT.EQ.2) XI=AN(1)
      GO TO 100
   50 CALL GLQUD(AUX,ANS)
      ANK=ANS
      IF(NOPT.EQ.1 .AND. DABS(TT-TTP).LE.0.001D0) XI=1.0D0
      IF(NOPT.EQ.1 .AND. DABS(TT-TTP).GT.0.001D0)
     1        XI=ANK*(TT-TTP)
  100 RETURN
      END
```

```
C
C     ............................................................
C
      SUBROUTINE GLQUD(AUX,ANS)
C     THIS SUBROUTINE SETS UP THE LAGUERRE-GAUSS INTEGRATION
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BBB/X(24),W(24)
      ANS=0.0D0
      S=0.0D0
      DO 10 I=1,24
      Y=X(I)
      CALL AUX(Y,Z)
   10 S=S+Z*W(I)
      ANS=S
      RETURN
      END
```

```
C
C     ..........................................................
C
      SUBROUTINE AUX(THET,FX)
C     THIS SUBROUTINE CALCULATES THE INTEGRAND OF P1,P2&P3
C     THE INTEGRAND OF P IS GIVEN BY A/B BELOW
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      COMMON/AAA/ KK
      IF(NB.EQ.1) GO TO 10
      THETK=THET+2.0D0*PI*(NB-KK)/NB
      A=TT*TTP*(TT-TTP*DCOS(THETK))+(TT*(THET*DSIN(THETK)+DCOS(THETK))
     1 -TTP)*EL**2
      B=(DSQRT(TT**2+TTP**2-2.0D0*TT*TTP*DCOS(THETK)+THET**2*EL**2))**3
     1 *DSQRT(EL**2+TTP**2)
      GO TO 20
   10 A=TT*TTP*(TT-TTP*DCOS(THET))+(TT*(THET*DSIN(THET)+DCOS(THET))-TTP)
     1 *EL**2
      B=(DSQRT(TT**2+TTP**2-2.0D0*TT*TTP*DCOS(THET)+THET**2*EL**2))**3
     1 *DSQRT(EL**2+TTP**2)
   20 C=A/B
      FX=DEXP(THET)*C
      RETURN
      END
```

```
C
C     ............................................................
C
      SUBROUTINE SPLCOE(XP,YP,M,CC)
C     THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR SPLINE
C     INTERPOLATION AS WELL AS INTEGRATION
C     THE INPUT PARAMETERS IN THE ARGUMENT ARE:
C     XP       VECTOR CONTAINING INDEPENDENT VARIABLE
C     YP       VECTOR CONTAINING DEPENDENT VARIABLE
C     M        NUMBER OF DATA POINTS
C     THE OUTPUT PARAMETER IN THE ARGUMENT IS:
C     CC       COEFFICIENT OF SPLINES
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XP(81),YP(81),D(81),P(81),E(81),CC(4,81),
     1AE(81,3),B(81),Z(81)
      MM=M-1
      DO 10 I=1,MM
      D(I)=XP(I+1)-XP(I)
      P(I)=D(I)/6.
   10 E(I)=(YP(I+1)-YP(I))/D(I)
      DO 20 I=2,MM
   20 B(I)=E(I)-E(I-1)
      AE(1,1)=1.0D0
      AE(1,2)=-1.0D0-D(1)/D(2)
      AE(1,3)=D(1)/D(2)
      AE(2,3)=P(2)-P(1)*AE(1,3)
      AE(2,2)=2.0D0*(P(1)+P(2))-P(1)*AE(1,2)
      AE(2,3)=AE(2,3)/AE(2,2)
      B(2)=B(2)/AE(2,2)
      DO 30 I=3,MM
      AE(I,2)=2.*(P(I-1)+P(I))-P(I-1)*AE(I-1,3)
      B(I)=B(I)-P(I-1)*B(I-1)
      AE(I,3)=P(I)/AE(I,2)
   30 B(I)=B(I)/AE(I,2)
      QR=D(M-2)/D(M-1)
      AE(M,1)=1.0D0+QR+AE(M-2,3)
      AE(M,2)=-QR-AE(M,1)*AE(M-1,3)
      B(M)=B(M-2)-AE(M,1)*B(M-1)
      Z(M)=B(M)/AE(M,2)
      MN=M-2
      DO 40 I=1,MN
      K=M-I
   40 Z(K)=B(K)-AE(K,3)*Z(K+1)
      Z(1)=-AE(1,2)*Z(2)-AE(1,3)*Z(3)
      DO 50 K=1,MM
      QR=1.0D0/(6.0D0*D(K))
      CC(1,K)=Z(K)*QR
      CC(2,K)=Z(K+1)*QR
      CC(3,K)=YP(K)/D(K)-Z(K)*P(K)
   50 CC(4,K)=YP(K+1)/D(K)-Z(K+1)*P(K)
      RETURN
      END
```

```
C
C     ...........................................................
C
      SUBROUTINE GAUSS(CA,M1,RQ)
C     THIS SUBROUTINE SOLVES A SYSTEM OF EQUATIONS
C     BY MEANS OF GAUSS ELIMINATION METHOD
C     THE INPUT PARAMETERS IN THE ARGUMENT ARE:
C     CA        COEFFICIENT MATRIX ¢A|
C     M1        NUMBER OF EQUATIONS
C     RQ        KNOWN VECTOR IN THE MATRIX EQUATION
C     THE OUTPUT PARAMETERS IN THE ARGUMENT ARE:
C     RQ        SOLUTION VECTOR
C     IER       ERROR MESSAGE
C     EPS       TOLERANCE OF THE GAUSS ELIMINATION TECHNIQUE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AQ(81),RQ(9 ),CA(9,9)
      DATA EPS/0.5D0/
      IEP=1
      DO 10 IP=1,M1
      DO 10 JP=1,M1
      AQ(IEP)=CA(JP,IP)
      IEP=IEP+1
   10 CONTINUE
C
      IF(M1)240,240,20
   20 IER=0
      PIV=0.0D0
      M3=M1*M1
      M4=M1
      DO 40 L=1,M3
      TB=DABS(AQ(L))
      IF(TB-PIV)40,40,30
   30 PIV=TB
      I=L
   40 CONTINUE
      TOL=EPS*PIV
C
      LST=1
      DO 180 K=1,M1
      IF(PIV)240,240,50
   50 IF(IER)80,60,80
   60 IF(PIV-TOL)70,70,80
   70 IER=K-1
   80 PIVI=1.0D0/AQ(I)
      J=(I-1)/M1
      I=I-J*M1-K
      J=J+1-K
      DO 90 L=K,M4,M1
      LL=L+I
      TB=PIVI*RQ(LL)
      RQ(LL)=RQ(L)
   90 RQ(L)=TB
C
      IF(K-M1)100,190,190
```

```
C
  100 LEND=LST+M1-K
      IF(J)130,130,110
  110 II=J*M1
      DO 120 L=LST,LEND
      TB=AQ(L)
      LL=L+II
      AQ(L)=AQ(LL)
  120 AQ(LL)=TB
C
  130 DO 140 L=LST,M3,M1
      LL=L+I
      TB=PIVI*AQ(LL)
      AQ(LL)=AQ(L)
  140 AQ(L)=TB
C
      AQ(LST)=J
C
      PIV=0.0D0
      LST=LST+1
      J=0
      DO 170 II=LST,LEND
      PIVI=-AQ(II)
      IST=II+M1
      J=J+1
      DO 160 L=IST,M3,M1
      LL=L-J
      AQ(L)=AQ(L)+PIVI*AQ(LL)
      TB=DABS(AQ(L))
      IF(TB-PIV)160,160,150
  150 PIV=TB
      I=L
  160 CONTINUE
      DO 170 L=K,M4,M1
      LL=L+J
  170 RQ(LL)=RQ(LL)+PIVI*RQ(L)
  180 LST=LST+M1
C
C
  190 IF(M1-1)240,230,200
  200 IST=M3+M1
      LST=M1+1
      DO 220 I=2,M1
      II=LST-I
      IST=IST-LST
      L=IST-M1
      L=AQ(L)+0.5D0
      DO 220 J=II,M4,M1
      TB=RQ(J)
      LL=J
      DO 210 K=IST,M3,M1
      LL=LL+1
  210 TB=TB-AQ(K)*RQ(LL)
      K=J+L
```

```
      RQ(J)=RQ(K)
  220 RQ(K)=TB
  230 GO TO 250
C
  240 WRITE(IO,300)
  250 RETURN
  300 FORMAT(10X,'ERROR DUE TO INCORRECT NUMBER OF EQUATIONS(
     1            ZERO OR NEGATIVE'/)
      END
```

```
C              '
C      ............................................................
C
       SUBROUTINE CD230(IREN,ALPHA,X,W,CL,CD)
C      THIS SUBROUTINE CONTAIN THE LIFT AND DRAG AIRFOIL DATA
C      CURVFFITS FOR NACA 230XXAIRFOIL(SMOOTH)  WITH CORRECTIONS
C      FOR THE EFFECT OF THICKNESS TO CHORD RATIO
C
C      INPUT PARAMETERS ARE:
C      IREN     CONTROL PARAMETER
C      ALPHA    ANGLE OF ATTACK
C      X        SPANWISE LOCATION
C      W        RELATIVE VELOCITY
C      OUTPUT PARAMETERS ARE:
C      CL       LIFT COEFFICIENT
C      CD       DRAG COEFFICIENT
C
C
       IMPLICIT REAL*8 (A-H,O-Z)
       COMMON/CCC/P2,TT,TTP,PI,EL,NB
       COMMON/EEE/RB,TCR75,SLTCR,CI75
       COMMON/DDD/ RDC,WO,RHO
       DATA ACLOI,ASIP,CDBSI,CDOI/-1.2D0,15.D0,0.0215D0,0.0074D0/
       DATA RCL,SLI,LEXP,ASCLO/.25,.103,5,90.0/
       RENS=3000000.0
       A=ALPHA*RDC
       AALPHA=DABS(ALPHA)
C
C       .....CALCULATE THE EFFECT OF THICKNESS TO CHORD RATO....
C
       FR=X
       T=-SLTCR*(FR-.75)+TCR75
       IF(IREN.EQ.1) RENS=1000*W*CI75
       FTL=.9534+2.0311*T-9.843700*T**2
       FTD=(1.0+2.0*(T-0.18))*(1-0.009*(0.000001*RENS-3.0))
       AR=RB/CI75
C
C     .....CALCULATE LIFT CURVEFIT CONSTANTS ....
C
       IF(T.EQ.PRET) GO TO 20
       SLF=SLI*FTL
       BCLF=0.-SLF*ACLOI
       IF(RENS.GT.6000000.) RENS=6000000.
       ASFP=ASIP*(1+1.1111*(0.18-T))*(1-0.02083*(3-.000001*RENS))
       ASFN=ACLOI-ASFP
       CLSPF=BCLF+SLF*ASFP-RCL*(-55.3332*T**2+16.5332*T-0.1088)
      1+0.0275*(0.000001*RENS-3.)
       CLSNF=BCLF+SLF*ASFN+RCL*(-55.3332*T**2+16.5332*T-0.1088)
      1+0.0275*(0.0000001*RENS-3.)
       SLSP=CLSPF/(ASFP-ASCLO)
       SLSN=CLSNF/(ASFN+ASCLO)
       BCLPS=CLSPF-SLSP*ASFP
       BCLNS=CLSNF-SLSN*ASFN
C
```

```
C     ....CALCULATE LIFT COEFFICIENT....
C
  20  CONTINUE
      IF(A.LT.ASFN) GO TO 40
      IF(A.LT.0.) GO TO 45
      IF(T.LT.0.18.AND.A.LT.ASFP*(1+0.02*(0.000001*RENS-3))) GO TO 501
      IF(T.LT.0.24.AND.A.LT.ASFP) GO TO 601
      IF(T.GE.0.24.AND.A.LT.ASFP) GO TO 701
      IF(T.LT.0.18.AND.A.LT.20.) GO TO 502
      IF(T.LT.0.24.AND.A.LT.20.) GO TO 602
      IF(T.GE.0.24.AND.A.LT.20.) GO TO 702
C     IF(A.LT.20.) GO TO 55
C     IF(A.LT.24.) GO TO 57
C     IF(A.LT.46) GO TO 59
      GO TO 60
 40   CL=SLSN*A+BCLNS
      GO TO 80
   45 CL=SLF*A+BCLF+RCL*(-55.3*T**2+16.5*T-0.109)*(ABS(A/ASFN))**LEXP
     1+(0.0275*(0.000001*RENS-3.))*(ABS(A/ASFN))**LEXP
      GO TO 80
  501 CL=SLF*A+BCLF-RCL*(-55.3*T**2+16.5*T-0.109+0.018*(0.000001*
     1RENS-3))*(A/ASFP/(1+0.02*(0.000001*RENS-3)))**LEXP
     GO TO 80
  601 CL=SLF*A+BCLF-RCL*(-55.3*T**2+16.5*T-0.109)*(A/ASFP)**LEXP
     1+(0.0425*(0.000001*RENS-3.))*(A/ASFP)**LEXP
      GO TO 80
  701 CL=SLF*A+BCLF-RCL*(-55.3*T**2+16.5*T-0.109)*(A/ASFP)**LEXP
     1+(0.01*(0.000001*RENS-3.))*(A/ASFP)**LEXP
      GO TO 80
 502  CL=(CLSPF-1.0)*(A-20)/(ASFP*(1+0.02*(0.000001*RENS-3))-20)+1
      GO TO 80
 602  CL=(CLSPF-1.0)*(A-20)/(ASFP-20)+1.0
      GO TO 80
 702  CL=(CLSPF-1.0-0.016667*(0.000001*RENS-3))*(A-20)/(ASFP-20)+1.0
      GO TO 80
 60   CL=1.5557*SIN(2*ALPHA)
      GO TO 80
  80  CONTINUE
C
C     .....CALCULATE DRAG CURVEFIT CONSTANTS....
C
      IF(T.EQ.PRET) GO TO 90
      CDBSF=CDBSI
      DF=(CDBSF-CDOI)/(ASFP-ACLOI)**2
C      CDMAX=1.5
      CDMAX=1.11+AR*.0178
      DS=(CDMAX-CDBSF)/(1.57-ASFP/RDC)**2
C
C     .....CALCULATE DRAG COEFFICIENT.....
C
   90 IF(A.LT.(0.-ASFP)) GO TO 100
      IF(T.GE.0.18.AND.A.LT.ASFP) GO TO 110
      IF(T.LT..18.AND.A.LE.ASFP*(1+0.02*(0.000001*RENS-3))) GO TO 111
  100 CD=CDMAX-DS*(1.57-AALPHA)**2
```

```
      GO TO 130
  110 CD=CDOI*FTD+DF*(A-ACLOI)**2
      GO TO 130
  111 CD=CDOI*FTD+(CDBSF-CDOI)*(A-ACLOI)**2/(ASFP*(1+0.02*(0.000001*
     1RENS-3))-ACLOI)**2
      GO TO 130
  130 CONTINUE
      RETURN
      END
```

```
C
C
C     ............................................................
C
      SUBROUTINE CD44 (ALPHA,CD,CL,X,W)
C     THIS SUBROUTINE CONTAINS THE LIFT AND DRAG AIRFOIL DATA
C     CURVFITS FOR NACA 44XX AIRFOIL(SMOOTH) WITH CORRECTIONS
C     FOR THE EFFECT OF THICKNESS TO CHORD RATIO
C
C     .... VARIABLE DEFINITIONS ....
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      COMMON/DDD/ RDC,WO,RHO
      COMMON/EEE/ RB,TCR75,SLTCR,CI75
      DATA ACLOI,ASIP,CDBSI,CDOI,PRET/-4.D0,14.D0,.021D0,.0077D0,0.D0/
      DATA RCL,SLI,ASCLO,LEXP/.33D0,.0971D0,90.D0,5/
      A=ALPHA*RDC
      AALPHA=DABS(ALPHA)
C
C     ..... CALCULATE THE EFFECT OF THICKNESS TO CHORD RATIO .....
C
      FR=X
      T=-SLTCR*(FR-.75)+TCR75
      AR=RB/CI75
C
C     ... CALCULATE THE EFFECT OF THICKNESS TO CHORD RATIO ...
C
      FTL=-1.1329*T+1.2039
      FTD=1.
C
C     ... CALCULATE LIFT CURVFIT CONSTANTS ...
C
      IF(T.EQ.PRET) GO TO 20
      SLF=1./(1./(SLI*FTL)+18.24/AR)
      BCLI=0.-(SLI*FTL)*ACLOI
      BCLF=0.-SLF*ACLOI
      CLSPI=(SLI*FTL)*ASIP+BCLI-SCL
      ASFP=ASIP*(1.+1.1905*(.18-T))
      ASFN=ACLOI-ASFP
      CLSPF=BCLF+SLF*ASFP-RCL*(5.5555*T**2-6.8433*T+1.8027)
      CLSNF=BCLF+SLF*ASFN+RCL*(5.5555*T**2-6.8433*T+1.8027)
      SLSP=CLSPF/(ASFP-ASCLO)
      SLSN=CLSNF/(ASFN+ASCLO)
      BCLPS=CLSPF-SLSP*ASFP
      BCLNS=CLSNF-SLSN*ASFN
C
C     ... CALCULATE LIFT COEFFICIENT ...
C
   20 CONTINUE
      IF(A.LT.ASFN) GO TO 40
      IF(A.LT.0.) GO TO 45
      IF(A.LT.ASFP) GO TO 50
```

```
      GO TO 60
   40 CL=SLSN*A+BCLNS
      GO TO 80
   45 CL=SLF*A+BCLF+RCL*(5.556*T*T-6.8433*T+1.8027)*(DABS(A/ASFN))**LEXP
      GO TO 80
   50 CL=SLF*A+BCLF-RCL*(5.556*T*T-6.8433*T+1.8027)*(DABS(A/ASFP))**LEXP
C   50 CL=SLF*A+BCLF-RCL*(A/ASFP)**LEXP
      GO TO 80
   60 CL=SLSP*A+BCLPS
      GO TO 80
   80 CONTINUE
C
C      ... CALCULATE DRAG CURVFIT CONSTANTS ...
C
      IF(T.EQ.PRET) GO TO 90
      CDBSF=CDBSI
      DF=(CDBSF-CDOI)/ASFP**2
      CDMAX=1.11+.018*AR
      IF(AR.GT.50) CDMAX=2.
      DS=(CDMAX-CDBSF)/(1.57-ASFP/RDC)**2
C
C      ... CALCULATE DRAG COEFFICIENT ...
C
   90 IF(A.LT.(0.-ASFP)) GO TO 100
      IF(A.LT.ASFP) GO TO 110
  100 CD=CDMAX-DS*(1.57-AALPHA)**2
      GO TO 130
  110 CD=CDOI*FTD+DF*A**2
      GO TO 130
  130 CONTINUE
      PERT=T
C
      RETURN
      END
```

```
C
C     ..........................................................
C
      SUBROUTINE CTRWT(V,RHB,CTRQ)
C     THIS SUBROUTINE CALCULATES THE NEGATIVE TORQUE OF COUNTER-WEIGHT
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/DDD/ RDC,WO,RHO
      COMMON/OOO/ RA,B1,B2,B3,B4,SIP,N2,N3
      DIMENSION XZ(81), Q(81), YI(81)
      SIPP=SIP/RDC
      DIP=DCOS(SIPP)
      R1=RA-B4/2.0
      NP2=(N2-1)*3+1
      NP3=(N3-1)*5+1
      VEF=VEF*DIP
      ND=0
      DO 800 I=1,NP2
      ND=ND+1
      XZ(I)=(I-1)*(R1-RHB)/FLOAT(NP2-1)
      YI(I)=B1-(B1-B2)*XZ(I)/(R1-RHB)
      REFF=(XZ(I)+RHB)*DIP
      VR=DSQRT(VEF**2+REFF**2*WO**2)
      PHI=DATAN(VEF/REFF/WO)
      Q(I)=-.5DO*DCOS(PHI)*RHO*VR**2*YI(I)*REFF*DIP
  800 CONTINUE
      QI=Q(ND)
      ZI=XZ(ND)
      SUMQ1=0.0DO
      CALL SPLINT(NP2,XZ,Q,SUMQ1)
      XZ(1)=0.DO
      Q(1)=QI
      DO 830 J=2,NP3
      XZ(J)=(J-1)*R2/FLOAT(NP3-1)
      JJ=ND+J-1
      YI(JJ)=2.DO*DSQRT(B3*B3*(1-(XZ(J)-B4)**2/B4**2))
      REFF=(XZ(J)+R1)*DIP
      VR=DSQRT(VEF**2+REFF**2*WO**2)
      PHI=DATAN(VEF/REFF/WO)
      Q(J)=-.25DO*DCOS(PHI)*RHO*VR**2*YI(JJ)*REFF*DIP
  830 CONTINUE
      CALL SPLINT(NP3,XZ,Q,SUMQ1)
      CTRQ=SUMQ1
  750 CONTINUE
      RETURN
      END
```

```
C
C     .................................................
C
      SUBROUTINE FACTRS(SKP1,SKP2,IS,ZETA,NP,J,XHB,YI,YIBZ,YIAZ)
C     THIS SUBROUTINE CALCULATES ALL THE INDUCTION FACTORS NEEDED
C
C     THE INPUT PARAMETERS IN THE ARGUMENT ARE:
C     ZETA      RADIAL COORDINATES
C     NP        NUMBER OF POINTS USED IN THE SPLINE INTERPOLATION
C     J         INDEX REFERING TO THE CORRESPONDING POINT ON THE BLADE
C     THE OUTPUT PARAMETERS OUT THE ARGUMENT ARE:
C     YI        INDUCTION FACTOR
C     YIBZ      INDUCTION FACROR AT ZETP-DEL
C     YIAZ      INDUCTION FACROR AT ZETP+DEL
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SKP1(9),SKP2(9),ZETA(81),IS(81),AA(4,81),XZ(81)
     1,Q(81),YI(81)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      DEL1=0.000001D0
      IF(NB.EQ.1) GO TO 10
      TT=TTP
      NOPT=2
      CALL CALCI(P2,NOPT)
C     P2 IS THE INTEGRAL FROM 0 TO INFINITY OF THE INTEGRAND M
C     WHERE I2=(ZETA-ZETP)*INTEGRAL OF M
   10 NOPT=1
      NAB=0
      NCTR=0
      ICTR=0
C     INTERPOLATES VALUES OF I BETWEEN SKP1(J) & SKP2(J) WHERE
C     SKP1(J) & SKP2(J) ARE ALREADY READ IN FOR EACH ZETP LOCATION
      DO 20 I=1,NP
      TT=ZETA(I)
C     FIND THE VALUE OF DO LOOP PARAMETER FOR WHICH ZETA=ZETP
      IF(DABS(TTP-TT).LT.0.000001D0) ICTR=I
C     STORE THE INDUCTION FACTOR IN ARRAY YI
      IF(TT.LT.SKP1(J) .OR. TT.GT.SKP2(J)) CALL CALCI(XI,NOPT)
C     STORE WHICH INDUCTION FACTOR CALCULATIONS HAVE BEEN SKIPPED
      IF(TT.LT.SKP1(J) .OR. TT.GT.SKP2(J)) YI(I)=XI
      IF(TT.GE.SKP1(J) .AND. TT.LE.SKP2(J)) IS(I)=1
   20 CONTINUE
      DO 30 I=1,NP
      ICK=0
      TT=ZETA(I)
      IF(IS(I).NE.1 .AND. TT.GE.SKP1(J)-.085D0 .AND. TT.LE.SKP2(J)
     1+.085D0 .OR. I.EQ.ICTR) ICK=2
C     NAB IS THE NUMBER OF POINTS AT WHICH THE DIRECT CALCULATION OF
C     THE INDUCTION FACTORS HAVE BEEN PERFORMED
      IF(ICK.EQ.2) NAB=NAB+1
      IF(ICK.EQ.2) XZ(NAB)=TT
C     STORE THE COMPUTED INDUCTION FACTORS IN ARRAY Q
      IF(ICK.EQ.2 .AND. I.NE.ICTR) Q(NAB)=YI(I)
      IF(I.EQ.ICTR) Q(NAB)=1.0D0
   30 IF(I.EQ.ICTR) NCTR=NAB
```

```
C     USE THE SPLINE INTERPOLATION METHOD TO OBTAIN INDUCTION
C     FACTORS THAT WERE SKIPPED IN THE DIRECT CALCULATIONS
C     SPLCOE IS A COMPUTER SUBROUTINE WHICH GENERATES SPLINE
C     INTERPOLATION COEFFICIENTS.
      CALL SPLCOE(XZ,Q,NAB,AA)
      DO 40 I=1,NP
      TT=ZETA(I)
C     USE SPLINE INTERPOLATION FORMULA
      IF(IS(I).EQ.1 .AND. I.LT.ICTR) YI(I)=
     1 AA(1,NCTR-1)*(XZ(NCTR)-TT)**3
     2+AA(2,NCTR-1)*(TT-XZ(NCTR-1))**3
     3+AA(3,NCTR-1)*(XZ(NCTR)-TT)
     4+AA(4,NCTR-1)*(TT-XZ(NCTR-1))
      IF(I.EQ.ICTR) YI(I)=1.0D0
   40 IF(IS(I).EQ.1 .AND. I.GT.ICTR)
     1  YI(I)=AA(1,NCTR)*(XZ(NCTR+1)-TT)**3
     2       +AA(2,NCTR)*(TT-XZ(NCTR))**3
     3       +AA(3,NCTR)*(XZ(NCTR+1)-TT)
     4       +AA(4,NCTR)*(TT-XZ(NCTR))
      IF(TTP.GT.XHB+DEL1) TT=TTP-DEL1
      IF(TTP.GT.XHB+DEL1)
     1  YIBZ=AA(1,NCTR-1)*(XZ(NCTR)-TT)**3
     2      +AA(2,NCTR-1)*(TT-XZ(NCTR-1))**3
     3      +AA(3,NCTR-1)*(XZ(NCTR)-TT)
     4      +AA(4,NCTR-1)*(TT-XZ(NCTR-1))
      IF(1.0D0-TTP.GT.DEL1) TT=TTP+DEL1
      IF(1.0D0-TTP.GT.DEL1)
     1   YIAZ=AA(1,NCTR)*(XZ(NCTR+1)-TT)**3
     2       +AA(2,NCTR)*(TT-XZ(NCTR))**3
     3       +AA(3,NCTR)*(XZ(NCTR+1)-TT)
     4       +AA(4,NCTR)*(TT-XZ(NCTR))
C     ALL INDUCTION FACTORS HAVE NOW BEEN CALCULATED AND STORED
C     IN YI, YIBZ, YIAZ. YIBZ & YIAZ ARE THE INDUCTION FACTORS
C     AT ZETP-DEL AND ZETP+DEL RESPECTIVELY.
C
      RETURN
      END
```

```
C
C     .................................................
C
      SUBROUTINE SPLINT(ND,ZZ,C,SUM)
C      THIS SUBROUTINE COMPUTES A FINITE INTEGRAL
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ZZ(81),ZC(4,81),C(81)
      CALL SPLCOE(ZZ,C,ND,ZC)
      NN=ND-1
      SUM=0.D0
      DO 10 K=1,NN
   10 SUM=SUM+.25D0*ZC(1,K)*(ZZ(K+1)-ZZ(K))**4
     1        +.25D0*ZC(2,K)*(ZZ(K+1)-ZZ(K))**4
     2        +.50D0*ZC(3,K)*(ZZ(K+1)-ZZ(K))**2
     3        +.50D0*ZC(4,K)*(ZZ(K+1)-ZZ(K))**2
C      WRITE(7,100) SUM,NN
C      WRITE(7,200) (ZZ(I),C(I),I=1,NN)
C  100 FORMAT(F10.4,I10)
C  200 FORMAT(4F10.4)
      RETURN
      END
```

```
C
C     ..........................................
C
      SUBROUTINE TINGRL(NS,NP,ZETA,XHB,DEL1,YI,YIBZ,YIAZ,TINT)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CCC/P2,TT,TTP,PI,EL,NB
      DIMENSION ZETA(81),S(81),XZ(81),Q(81),YI(81)
      DEL=DEL1
      ND=0
      ARG=NS*PI*(TTP-XHB)/(1.D0-XHB)
      IF(TTP.LE.DEL+XHB) GO TO 30
C
C     THIS SECTION CALCULATES THE INTEGRAL FROM XHB TO ZETP-DEL
C
      DO 10 I=1,NP
      ND=ND+1
      IF(ZETA(I)-TTP.GE.-DEL1) GO TO 20
      XZ(I)=ZETA(I)
      ARGP=(ZETA(I)-XHB)/(1.0D0-XHB)
C     STORE THE INTEGRAND IN ARRAY Q
   10 Q(I)=DCOS(NS*PI*ARGP)*YI(I)/(ZETA(I)-TTP)
   20 XZ(ND)=TTP-DEL
      ARGP=(XZ(ND)-XHB)/(1.0D0-XHB)
      Q(ND)=DCOS(NS*PI*ARGP)*YIBZ/(XZ(ND)-TTP)
C     INTEGRATE Q USING SPLINE INTEGRATION FORMULA
      SUMQ=0.0D0
      CALL SPLINT(ND,XZ,Q,SUMQ)
   30 CONTINUE
      IF(TTP-XHB.LT.DEL .OR. 1.0D0-TTP.LT.DEL) DEL=DEL/2.0D0
C
C
C     THIS SECTION CALCULATES THE INTEGRAL FROM ZETP-DEL TO ZETP+DEL
C
      SUMR=0.0D0
      ATERM=0.D0
      BTERM=0.D0
      IF(NB.EQ.1) GO TO 40
      ATERM=DCOS(ARG)*2.0D0*DEL*P2
   40 BTERM=-NS*PI*2.0D0*DEL*DSIN(ARG)
      SUMR=ATERM+BTERM
      DEL=DEL1
      IF(1.0D0-TTP.LE.DEL) GO TO 60
C
C     THIS SECTION CALCULATES THE INTEGRAL FROM ZETP+DEL TO 1
C
      ND=1
      XZ(ND)=TTP+DEL
      ARGP=(XZ(ND)-XHB)/(1.0D0-XHB)
      S(ND)=DCOS(NS*PI*ARGP)*YIAZ/(XZ(ND)-TTP)
      DO 50 I=1,NP
      IF (ZETA(I)-TTP.LE.DEL1) GO TO 50
      ND=ND+1
      XZ(ND)=ZETA(I)
C     STORE THE INTEGRAND IN ARRAY S
```

```
      ARGP=(XZ(ND)-XHB)/(1.0D0-XHB)
      S(ND)=DCOS(NS*PI*ARGP)*YI(I)/(XZ(ND)-TTP)
   50 CONTINUE
C     INTEGRATE S USING SPLINE INTEGRATION FORMULA
      SUMS=0.0D0
      CALL SPLINT(ND,XZ,S,SUMS)
C     TINT IS THE TOTAL INTEGRAL FROM XHB TO 1
   60 TINT=SUMQ+SUMR+SUMS
      RETURN
      END
```

```
C
C     .............................................................
C
      SUBROUTINE SKIP(ZETP,RB,OMEGA,VEL,N,SKP1,SKP2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ZETP(9),SKP1(9),SKP2(9)
      WRITE(7,501) VEL
  501 FORMAT( '  HERHE VEL ',F10.4)
      WRITE(7,502) (ZETP(I),I=1,N)
  502 FORMAT( '  ZETAP VAL ',F10.4)
      SKP1(1)=ZETP(1)
      SKP2(N)=1.0D0
      FR=RB*OMEGA/VEL*0.0020973D0
      IF(FR.GT.1.) FR=1.0
      DO 50 J=2,N
      SKP1(J)=ZETP(J)-0.050-0.025*ZETP(J)**2-.025*ZETP(J)**2*FR**3
      IF(SKP1(J).LT.ZETP(1)) SKP1(J)= ZETP(1)+0.0001
   50 CONTINUE
      DO 100 J=1,N-1
      SKP2(J)=ZETP(J)+0.050+.025*(ZETP(N-1))**2
     1       +.025*(ZETP(J)/ZETP(N-1))**2*FR**3
      IF(SKP2(J).GT.ZETP(N)) SKP2(J)= ZETP(N)-0.0001
  100 CONTINUE
      WRITE(7,200) (SKP1(I),SKP2(I),I=1,N)
  200 FORMAT(2F12.6)
      RETURN
      END
```

1 Report No	2 Government Accession No	3 Recipient's Catalog No
NASA CR-174921		

4 Title and Subtitle	5 Report Date
Aerodynamic Analysis of a Horizontal Axis Wind Turbine by Use of Helical Vortex Theory Volume II: Computer Program Users Manual	April 1985
	6 Performing Organization Code
7 Author(s)	8 Performing Organization Report No
T.G. Keith, Jr., A.A. Afjeh, D.R. Jeng and J.A. White	776-33-41
	10 Work Unit No
9 Performing Organization Name and Address	11 Contract or Grant No
The University of Toledo Toledo, Ohio	NCC 3-5
	13 Type of Report and Period Covered
12 Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Wind Energy Technology Division Washington, D.C. 20546	14 Sponsoring Agency ~~Code~~ Report No. DOE/NASA/0005-2

15 Supplementary Notes

Final report. Prepared under Interagency Agreement DE-AI01-76ET20320. Project Manager, Robert Corrigan, Wind Energy Project Office, NASA Lewis Research Center, Cleveland, Ohio 44135.

16 Abstract

A description of a computer program entitled VORTEX that may be used to determine the aerodynamic performance of horizontal axis wind turbines is given. The computer code implements a vortex method from finite span wing theory and determines the induced velocity at the rotor disk by integrating the Biot-Savart law. It is assumed that the trailing helical vortex filaments form a wake of constant diameter (the rigid wake assumption) and travel downstream at the free stream velocity. The program can handle rotors having any number of blades which may be arbitrarily shaped and twisted. Many numerical details associated with the program are presented. A complete listing of the program is provided and all program variables are defined. An example problem illustrating input and output characteristics is solved.

17 Key Words (Suggested by Author(s))	18 Distribution Statement
Horizontal axis wind turbine, Helical vortex theory	Unclassified - unlimited STAR Category 44 DOE Category UC-60

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of pages	22 Price*
Unclassified	Unclassified	85	A05

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document