FREE WAKE TECHNIQUES FOR ROTOR AERODYNAMIC ANALYSIS: VOL II: VORTEX SHEET MODELS

NASA CR-166,435

A. TANUNIDJAJA

NASA-CR-166435
19830011441

A Reproduced Copy

OF

Reproduced for NASA

by the

NASA Scientific and Technical Information Facility

NASA CONTRACTOR REPORT 166435

(NASA-CR-166435) FREE WAKE TECHNIQUES FOR ROTOR AERODYNAMIC ANALYLIS. VOLUME 2: VORTEX SHEET MODELS (Massachusetts Inst. of Tech.) 162 p HC A08/MF A01 CSCL 01A N83-19712 Unclas G3/02 09170

Free Wake Techniques for Rotor Aerodynamic Analysis; Volume No. II-Vortex Sheet Models

A. Tanuwidjaja

CONTRACT NAG2-47
December 1982

NASA

N83-19712 #

NASA CONTRACTOR REPORT 166435

Free Wake Techniques for Rotor Aerodynamic Analysis; Volume No. II - Vortex Sheet Models

A. Tanuwidjaja
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts

Prepared for Ames Research Center under
NASA Cooperative Agreement No. NAG2-47

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TABLE OF CONTENTS

NOMENCLATURE

α	Half size of the vortex segment
ϵ_1	Thickness of the sheet element
ϵ_2	Core radius of the segment element
$\frac{\Gamma}{\pi R^2}$	Bound circulation distribution
ϕ	Azimuthal position of vortices starting at the blade ($\phi = 0$)
η	Dimensionless spanswise position along the blade (r/R)
σ	Solidity of rotor
k	Parameter for the distribution of vorticity between the tip and the most outboard sheet vortex; the root and the most inboard sheet vortex.
L_T	Equivalent to C_T, thrust coefficient - $C_T = \frac{T}{\rho\pi\Omega^3 R^4}$
L_Q	Equivalent to C_Q, torque coefficient - $C_Q = \frac{P}{\rho\pi\Omega^3 R^5}$
r	Radial coordinate
R	Rotor radius
R_T	Radial position of tip vortex
T	Thrust of rotor
P	Power of rotor
W_s	Lamb vortex ring self-induced velocity

ABSTRACT

In the free wake analysis of a rotor, velocities are computed everywhere in the wake and integrated to determine the wake displacement. Newly developed techniques of free wake analysis of a hovering rotor are presented. These techniques are designed to predict better wake structures and, thus, blade circulation distributions. Previous analyses have been modified to include the roll up process according to the Betz criteria. The near wake is modelled in four different ways, resulting in four different vortex sheet models. The results show that the azimuthal location of the starting point of the roll up process is important. The computational results are compared to the most recent available experimental data.

1. INTRODUCTION

There exists a need for an efficient computational or analytical tool for designing hovering helicopter rotors, as well as predicting the aerodynamics of these rotors. The need for a free wake model is well recognized (references 1, 2, 4 and 5). In reference 1, the wake is modelled by a finite number of discrete vortex filaments; by contrast, the model in reference 2 assumes a continuous vortex sheet.

The model in reference 2 and its associated computer code was originally developed to analyze the flow through a wind turbine. More recently, this model and the resulting modified computer codes have been applied to the study and understanding of rotor hovering aerodynamics (3). The working air is assumed to be inviscid and incompressible. Each rotor blade is represented by a lifting line on which the bound vorticity is assumed to concentrate. However, the effect of "spilling" of air around the blade tip (tip loss) has been neglected.

The entire spiral wake system is divided into 3 sections. These are: (I) the near wake, (II) the intermediate wake, (III) the far wake, as shown in figure A-1. All the wakes contain vortex sheets, tip and root vortices. The near wake usually extends from the blade to an azimuth position which is a parameter of the model, but is usually around 90 degrees. There is a parameter K which determines the distribution of vorticity at the most outboard station between the sheet and tip vortex, and also at the most inboard sheet and the root vortex. The mesh size of vortex sheets in the near wake depends on the spacing of the blade stations (or nodes). These stations further subdivide the blade into sections. A second set of points on the blade, called centers, is defined at the middle of each section. The intermediate wake has a coarser mesh size and extends to the start of the far wake. Therefore, an overlapping region called the transition wake, between the near and intermediate wake, is necessary. The far wake is modelled by constant radii vortex cylinders in which the radii are determined by the final spanswise spacing of the intermediate mesh. The vertical position of this vortex cylinder is located at one vortex spacing below the last intermediate wake vortex. All the induced velocities are computed by integration of the appropriate form of the Biot- Savart Law. The far wake contributions to the induced velocities in the near and intermediate wakes are obtained by series solution techniques. Self induced effects are included by the method described in Appendix I.

The induced velocities cannot be computed at the points (nodes) which define the elements, such as the two end points of a segment element or the four corner points of a sheet element. This is inherent in the use of discrete vortex sheet elements of finite size as described above. Thus, new positions on the wake, called centers, need to be defined. At these points the induced velocities in the wake are calculated. Finally, the velocity involved at a node is obtained by linear interpolation of induced velocities on the four centers around the node.

The computational procedure has to be an iterative one, as the determination of the induced velocity requires the knowledge of the location of vorticity, which is not known a priori. Consequently, the results of the semi-rigid wake have been used to initiate the calculation in the free wake analysis. The basic assumption of the semi-rigid wake is that the velocity of a vortex element is constant with respect to time and equal to the velocity through that point in the rotational plane on which it originated. Another approximation is made by assuming the angle of inclination of the vortex element is constant with respect to time.

Some difficulties are encountered in the use of this vortex sheet model; for instance, the edge singularities do not cancel between sheets, because of their relative displacements as the wake distorts. In addition, as the intermediate wake goes beyond 280 degrees, the model may not give a convergent solution, due to the modelling of the roll up process. Finally, there is a problem with the root vortex which will be further elucidated in the next section. Thus there is a need for an improvement in the free wake model to study the development of powered rotor aerodynamics.

The results from these two models in references 1 and 2 compare favorably with the available experimental data; however, the computational techniques used are time consuming and may, on occasion, not give a convergent solution for reasons indicated above.

However, the simpler model proposed in reference 6 requires only the determination of induced velocity at the blades, which are represented by lifting lines, and at the infinite vortex lines located below the blade. It gives results which compared well with the available test data and also yields useful insight into the aerodynamics of rotating wings.

In this modified model, it is assumed that a roll up of vortex sheets into isolated vortex filaments occurs in the intermediate wake. Because these rolled up vortices are discrete vortex filaments, the presence of a transition wake is not necessary. A tip loss of .985 as used in reference 8 has been assumed. A more efficient technique involving the use of elliptical integrals for the induced velocity computations due to the vortex cylinder has been adopted. This technique is described in (6). The azimuthal position at which the vortices roll up has been retained as a parameter than can be varied. The near wake is modelled in four different ways, resulting in four different vortex sheet models which will be described in detail in the next section.

II. MODELLING OF THE WAKES AND DISCUSSION OF RESULTS

Summary of Free Wake Models

MODEL 1 Essentially the same as ref. 2, in which the near and intermediate wakes are represented by a segmented vortex sheet connecting a tip and root vortex. The far wake is represented by vortex cylinders as in ref. 6.

MODEL 2 The near wake is treated as in Model 1 until an azimuth of $\phi = 255°$, after which it is rolled up into three vortex filaments as in ref. 6. The location of these filaments at $\phi = 255°$ is determined as if the filaments had rolled up immediately at $\phi = 0$.

MODEL 3 Same as Model 2, except that the near wake is rolled up before first encounter, at $\phi = 45°$, and is assumed to remain in the rotor plane until roll up. The geometry of the rolled up vortices at $\phi = 45°$ is determined as in Model 2. This avoids the lengthy computations required to establish the geometry of the vortex sheets and hence is computationally much more efficient than Model 2.

MODEL 4 Same as Model 3, except that the full vortex sheet model, including tip and root vortices, is used to establish the geometry of the vortex filaments at roll up.

First Vortex Sheet Model

This model is the original formulation of ref. 2, as explained above and in the Introduction. A complete description is contained in the appendix and in ref. 2. The following paragraphs will discuss the exploratory results obtained by varying a wide range of parameters, such as viscous core size, distribution of vorticity at the most outboard station between the vortex sheet and the tip vortex and the effects of the root vortex. The computed wake geometry and the rotor thrust coefficients are compared with the experimental data of reference 4. A brief summary of results can be explained as follows:

1) No important effects are caused by increasing the core depth of the vortex sheets, but the coefficient of thrust (C_T) is increased and the vertical location of the tip vortex (Z_T) is decreased by enlarging the core radius of the vortex filaments. (See Figure 2.)

2) Suppressing the strength of the root vortex to zero increases the contraction of the tip vortex. Z_T is also increased, but C_T is decreased - all slightly. (See Figure 3.) In general, the root vortex has negligible effects on the solution, as was also shown in ref. 6.

3) Figure 3 shows the effect of increasing the azimuth of the intermediate wake from 260 to 360 degrees. Note that the Z_T location for the 360 degree case starts to diverge from the experimental solution. However, at 260 degrees azimuth, the 360 degree case shows better agreement with the experimental data.

Further study has been made to extend the intermediate wake beyond 360 degrees, but converged results could not be obtained. One of the reasons was the migration of the root vortex up through the rotor. As the spacing of the constant radius vortex cylinder approaches zero, the induced velocity approaches infinity, causing convergence problems even though the vertical velocity and circulation near the root approach zero. As discussed above, when the strength of the root vortex is suppressed to zero, no important effects either in Z_T, C_T, or converged solution are obtained and convergence is improved.

Another effect causing lack of convergence was probably due to the roll up of the tip vortex predicted by the sheet modelling of the intermediate wake. Decreasing the mesh size of the intermediate wake could improve the modelling of the roll up process. Since a finer mesh size of the wake would result in excessive computational time, and since the parallel study of reference 6 has indicated satisfactory results with a fast roll up while avoiding the convergence problems, the free wake program was modified to include the roll up of the intermediate wake, according to the Betz criteria explained in detail in reference 6. This led to the second vortex sheet model discussed next.

Second Vortex Sheet Model

In this model, the near and the far wakes are again represented as described in reference 2. The near wake is extended to 255 degree azimuth. The roll up of the sheet vortices is assumed to take place at the intermediate wake. The roll up schedule as discussed in reference 5 has been adopted. Because these rolled up vortices are isolated vortex filaments, the existence of a transition wake is not necessary. These rolled up vortices adjoin the near wake and extend to 720 degrees.

In this modified model there are three rolled up vortices. They are the tip, center and root vortices. It is logical to roll up the vorticity from near the tip (allowing for a tip loss) to the peak of circulation to produce the tip vortex. Then, the center vortex is obtained by rolling up the vorticity from the peak circulation to about 80% span where $d\Gamma/d\eta$ is minimum. The root vortex contains the vorticity from about 80% span to the last station of the blade and is kept for this case of three vortex filaments.

The determination of the induced velocity due to the presence of these vortices requires the knowledge of their positions, especially the starting positions, in the intermediate wake region. However, their starting positions are not known a priori; consequently a procedure has to be developed to compute them. We have noted in the above that the vortex sheet associated with a particular segment of bound vorticity (there are three such segments) on the blade is subdivided into several vortex sheet elements of finite size for computational purposes. The centroid of the rolled up vortices are computed at each ϕ, according to the Betz roll up criteria starting from $\phi = 0$, and the velocity computed at these points due to the entire wake, consisting of the bound circulation, the vortex filaments including self induced effects, the rolled up intermediate wake vortices and the vortex cylinders representing the far wake. It must be emphasized that in the calculation of the induced velocity on the blades, the near wake region is taken to be a continuous vortex sheet rather than a few vortex filaments as described above.

The computational results are shown in figures 4, 5 and 6, compared to the test data of reference 8. Figure 4 shows the effect of the parameter K on the bound circulation and location of the tip vortex. When K equals zero the tip and root vortices have zero strength; thus the near wake is represented only by vortex sheet elements. However, when K equals 1, the strength of tip vortex is the same as the amount of circulation at the last center of the blade, whereas the strength of the root vortex is equal to the value of the bound circulation at the first center of the blade and the most outboard and inboard vortex sheets have zero velocity. K equal to .5 means that the distribution of vorticity is divided equally between the tip vortex and the most outboard vortex sheet, and the root vortex and the most inboard vortex sheet. Note that the vortex locations shown at the first encounter with the blade are those of the rolled up vortices although, in computing blade loads, the full vortex sheet system is used since roll up does not occur until 255°.

Figure 5 describes the behavior of the roll up process. The result shown is for K equal to 1. The interaction of the tip vortex with the most outboard sheet vortices are studied for the first 50 degree azimuth. The most outboard sheet vortices and the intermediate wake tip rolled up vortices induce upward and inward velocity components on the tip vortex, resulting in its migration up through the rotor. In turn, the tip vortex affects the sheet vortices in the opposite manner. Because of the upward displacement of the tip vortex in the starting region of the near wake, its vertical location at the first blade/vortex encounter is slightly above the test value. Therefore, the velocity induced at the blade (80% span), especially the vertical component (down wash), yields lower bound circulation than the predicted value. This roll up behavior causes slow convergence of the computer program. Furthermore, because of the coarse grid size needed to keep computational times manageable, the roll up procedure is probably only an approximate model of the actual roll up. This point is discussed further in reference 11.

The results shown are in a reasonable agreement with the test data. From the discussion above, this model suggests that the roll up of the sheet vortices should have been assumed earlier than 255 degree azimuth as indicated by figure 6, where the near wake is extended only to 40 degree azimuth, resulting in a better agreement between the experimental and computed circulation distribution. This conclusion leads to the third vortex sheet model.

Third Vortex Sheet Model

The near wake representation of the second vortex sheet model is modified to reduce the computational time. The near wake is extended only to 40 degrees. Because of its negligible distortion over this distance the vertical displacements of the vortices in this wake are assumed to be zero. This implies that the near wake is positioned on the rotational plane of the rotor. The results of the semi-rigid wake have been used to determine the spanwise positions of these vortices. Therefore, the determination of induced velocities at this near wake is not necessary. The starting locations of the intermediate wake rolled up vortices are treated in the same manner as in the second vortex sheet model. It must be emphasized that the continuous vortex sheet representation of the near wake is now used only to determine the induced velocity at the rotor blade from which it springs.

The results using this model are shown in figures 7, 8 and 9. The number of rolled up vortices is three and the K parameter is equal to 1. Figures 7 and 8 compare the results for a two bladed and a four bladed rotor with the experimental data of reference 4. Lack of test data on the blade circulation distribution and the wake geometry for a four bladed rotor does not enable one to compare results, except for the values of C_T/σ and C_Q/σ. As the number of rotor blades increases, the effect of first blade/vortex interaction becomes more important. An improvement of the blade representation is necessary to yield better results. Reference 7 shows the results using lifting surface theory, and these indeed are better than the lifting line solution for a four bladed rotor.

Another result for a two bladed rotor is compared with the test data of reference 8 in Figure 9. The predicted value of peak circulation is slightly low ($\frac{\Gamma}{\Omega R^2} = .0209$). This is caused by the relative vertical location predicted for the rolled up tip vortex at its first encounter with the blade compared to test data. Thus, an improvement of this vortex sheet model is necessary to predict the tip vortex location more accurately, leading to the fourth model.

Fourth Vortex Sheet Model

In this model the displacement of the rolled up vortices at time of roll up is determined, as in the previous models, by computing the velocities at the location of a hypothetical vortex filament assumed to roll up starting at $\phi = 0$ in the near wake. However, in the two previous models, the effect of the tip vortex in the near wake on the tip roll up is essentially neglected. This is because the induced velocities on the rolled up vortex lines due to the continuous vortex sheet in the near wake has been excluded in determining their geometry, and the induced effect of the vortex filaments on each other and themselves used instead. In this modified model the full vortex sheet, including tip and root vortices, is used and not the vortex filament. It is also assumed that the vertical position of the continuous vortex sheet representation of the near wake is equal to the vertical position of the tip vortex line. Consequently, this vortex sheet induces only vertical velocity on the vortex lines. The spanwise positions of the near wake vortices are assumed to be constant radii vortex filaments and sheets, of which the radii are the stream lines originating from the blade. Therefore, the determination of the induced velocities in the near wake is not necessary, except at the vortex filaments. The vortex core size becomes critical because of the close spanswise positions between the tip vortex and the tip vortex line. The computational results of this model are shown in figures 10 to 16.

In figure 10, the number of rolled up vortices is four. The roll up schedule is as follows: from the peak of circulation to the near tip region, the vorticity is rolled up into the tip vortex; from the peak of circulation to 79% span, the roll up process yields the first center vortex; then from the 70% to the 35% span, the roll up produces the second center vortex; finally, from 35% span to the last center of the blade, the vorticity is rolled up into the root vortex. The strength of the root vortex has been suppressed, for this case only, to zero in order to avoid the convergence problem noted above. Fig. 10 shows that the effect of varying the azimuthal size of the near wake indicates that this is not a highly sensitive parameter.

Figure 11 shows the effect of different roll up schedules for the center rolled up vortices only. The dashed curve has the new roll up schedule as follows: the first center vortex rolls up the vorticity from the peak of circulation to the 82.5% span; the second center vortex contains the vorticity from 82.5% to 35% span; the root and tip roll up vortices have the same roll up schedule as described above.

Figures 12 to 15 compare the results with test data of reference 4. The new roll up schedule (above) is used. Finally, figure 16 shows the effect of varying the number of rolled up vortices from four to the original three used in previous models and of varying the slope of the lift curve from 2π to 6.15.

Summary of Free Wake Analysis

The free wake analysis may be summarized as follows:

1) The rotor blade is treated as a lifting line.

2) The fluid is assumed to be inviscid and incompressible. A vortex core is used to avoid the singularities.

3) The wake is divided into three sections, a near wake attached to the blade, an intermediate wake, and a far wake.

4) Depending on the representation of the near wake, different vortex sheet models can be proposed.

5) Except in the first vortex sheet model, the intermediate wake is assumed to have rolled up. In the second vortex sheet model, the vortex sheets are rolled up after the first blade/vortex interaction. In the third and fourth sheet models, the roll up process takes place before the first blade encounter. The roll up process is based on Betz theory of conservation of momentum.

6) A peak bound circulation is expected near the tip due to the close encounter between the blade and the tip vortex from the preceeding blade and this determines the strength of the tip vortex.

7) The intermediate wake is rolled up into either three or four vortex filaments. The roll up schedule is explained in section 4. The root vortex contribution to the circulation distribution can be neglected as shown in reference 6.

8) The far wake is represented by semi-infinite vortex cylinders, starting at a vertical distance from the rotor one vortex spacing below the last intermediate wake vortex. The induced velocities from these vortex cylinders are obtained from the solutions in ref. 6.

9) The slope life curve used is 2π and the convergence criteria is 3% for circulation and 6% for wake geometry. A tip loss of .985 is used.

10) The intermediate wake is extended to 720° for a two bladed rotor and 360° for a four bladed rotor.

III. CONCLUSIONS AND RECOMMENDATIONS

This report presents newly developed analytical techniques of free wake analysis in a hovering rotor. Results are shown for the locations of the tip vortex and the bound circulation distribution.

These results indicate the sensitivity of the blade loading distribution to the location of the tip vortex at the first encounter. Investigation of different vortex sheet models led to two important conclusions. These are:

(I) the roll up of the vortex sheet has to be included;

(II) the knowledge of the starting point of this roll up process is important.

These have been studied and yield different wake geometries (especially the tip vortex location at first encounter) for various vortex sheet models.

These methods have demonstrated the ability to predict vortex wake geometry and performance of a two and four bladed rotor satisfactorily.

Recommendations for further investigations include:

1) Determine the starting point of the roll up process.
2) Develop more efficient computational techniques.
3) Use a lifting surface representation of the rotor blade.
4) Obtain experimental data on wake geometry and on blade circulation distribution for four or more bladed rotors.

References

1. Scully, M., "Computation of Helicopter Rotor Wake Geometry and Its Influence on Rotor Harmonic Airloads," MIT ASRL TR 178-1, March 1975.

2. Gohard, J.D., "Free Wake Analysis of Wind Turbine Aerodynamics," MIT ASRL TR 184-14, September 1978.

3. Chung, S.Y., and M. Brower, Private communication.

4. Landgrebe, A.J., "An Analytical and Experimental Investigation of Helicopter Rotor Hover Performance and Wake Geometry Characteristics," USAAMRDL TR71-24, June 1971.

5. Kocurek, J.D., L.F. Berkowitz and F.D. Harris, "Hover Performance Methodology at Bell Textron," AHS Preprint 80-3, May 1980.

6. Miller, R.H., "Simplified Free Wake Analysis for Rotors," FFA (Sweden) TN 1982-7; also MIT ASRL TR 194-3.

7. Miller, R.H., "Application of Fast Free Wake Analysis Techniques to Rotors," 8th European Rotorcraft Forum, paper No. 2.4, August 31 through September 3, 1982.

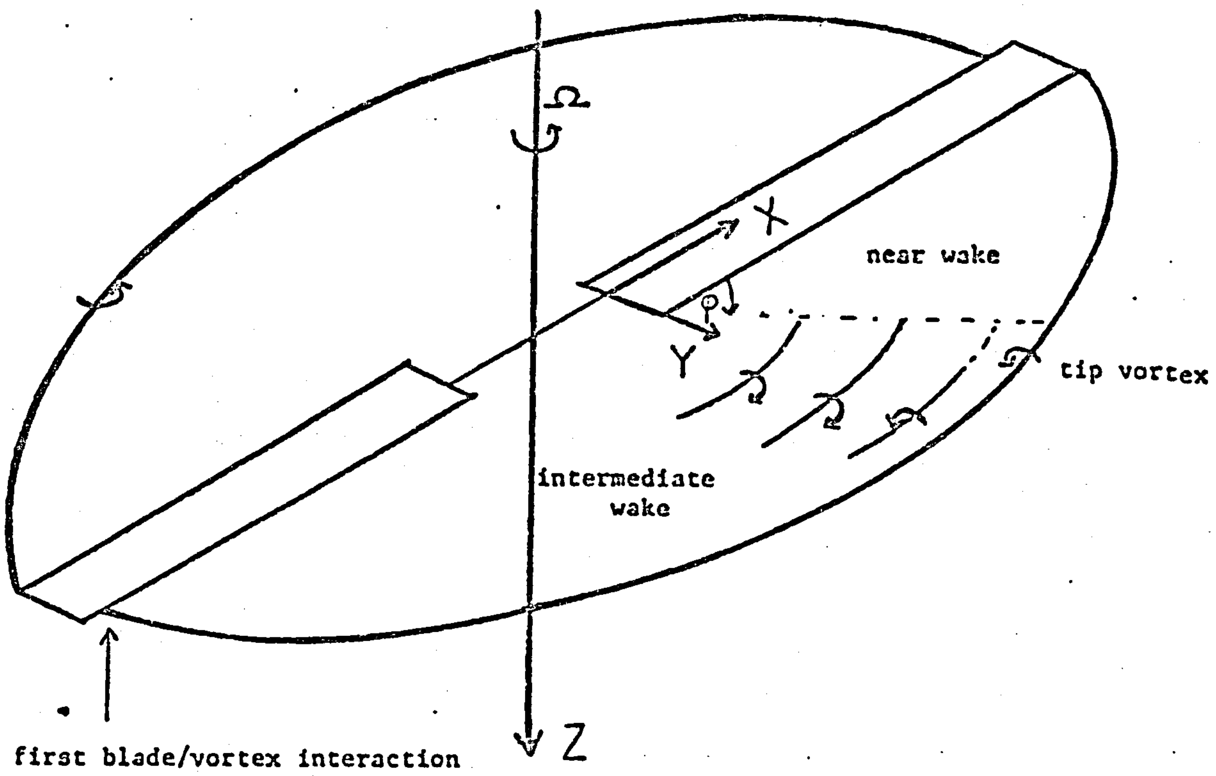
8. Johnson, W., "Comparison of Calculated and Measured Model Rotor Loading and Wake Geometry," NASA TM 81189.

9. Lamb, H., HYDRODYNAMICS, 6th Edition, Dover, 1932, p. 241.

10. Miller, R.H., "Free Wake Techniques for Rotor Aerodynamic Analysis, Volume I: Summary of Results and Background Theory,"

11. Brower, M., "Free Wake Techniques for Rotor Aerodynamic Analysis, Volume III: Vortex Filament Models,"

A)



B)

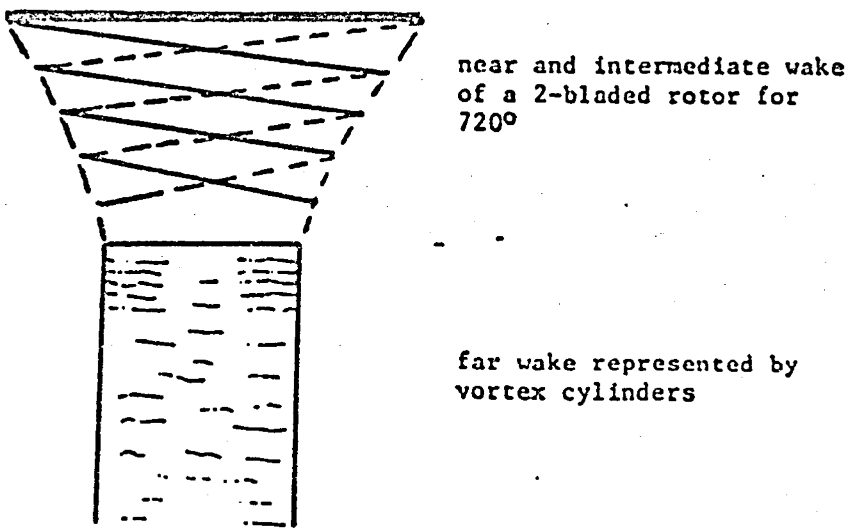


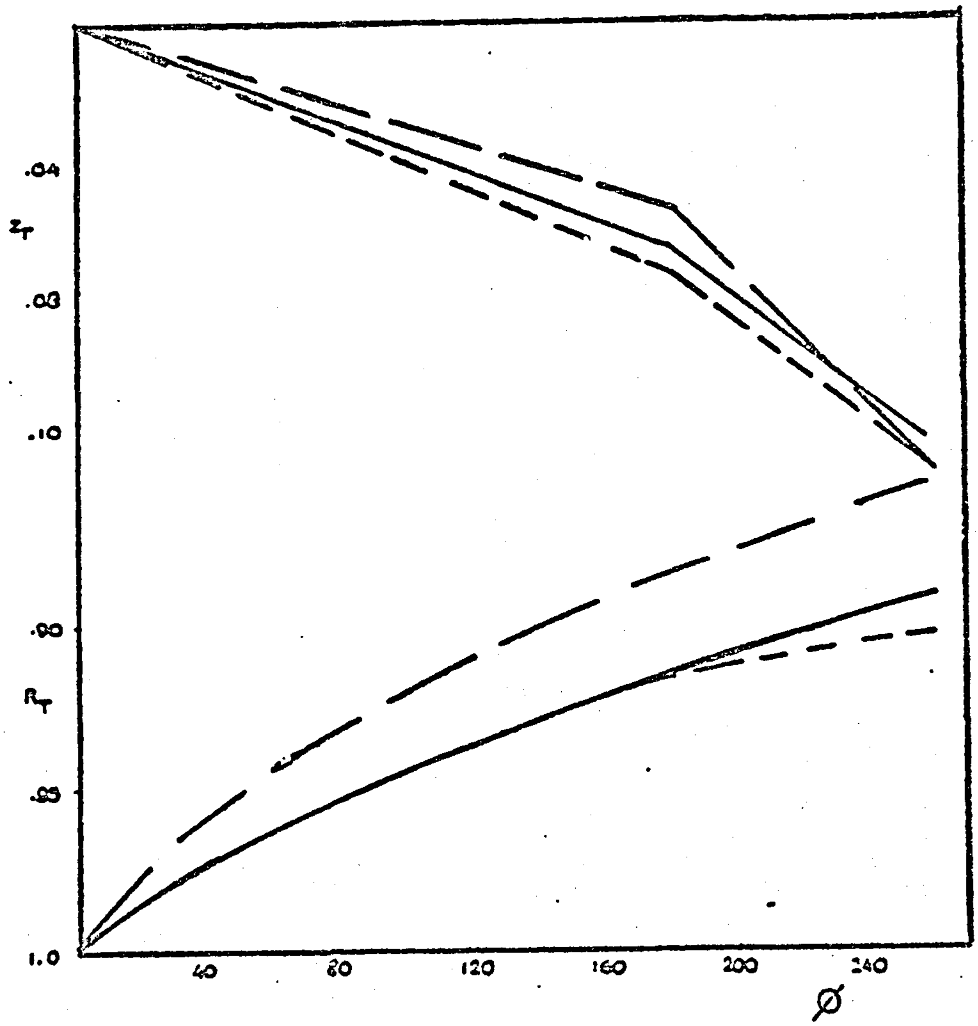
Figure A-1. A) Plan view, B) side view

Fig. 1: Effect of varying core size of tip vortex, ϵ_2. Untwisted blade.

——— ϵ_2 = .05 C_T = .0027

——— ϵ_2 = .01 C_T = .00255

— — Ref. 4 Test C_T = .0028



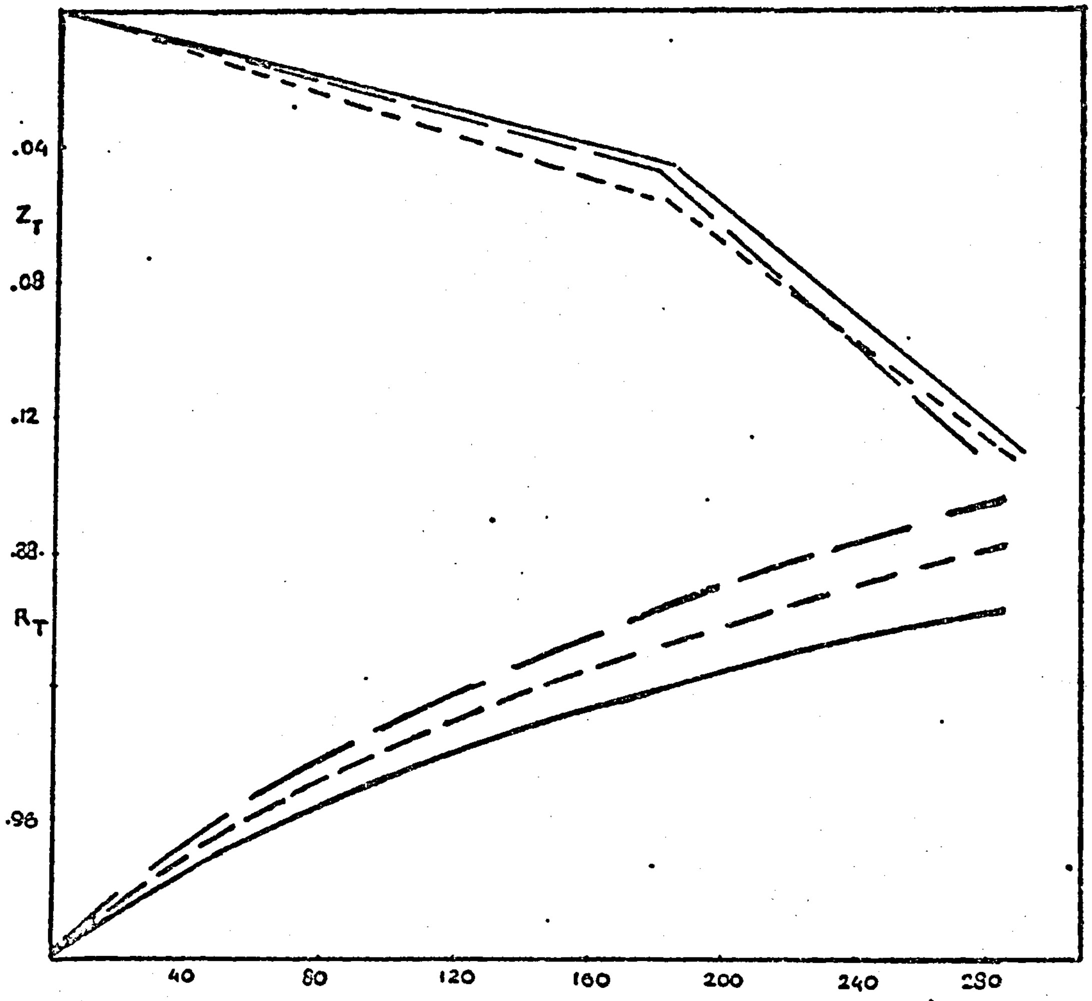


Fig. 2: Effect of neglecting effect of root vortex. Twisted blade.

———	Root vortex strength ≠ 0	C_T = .00281
- - - -	Root vortex strength = 0	C_T = .00275
— —	Ref. 4 test	C_T = .0028

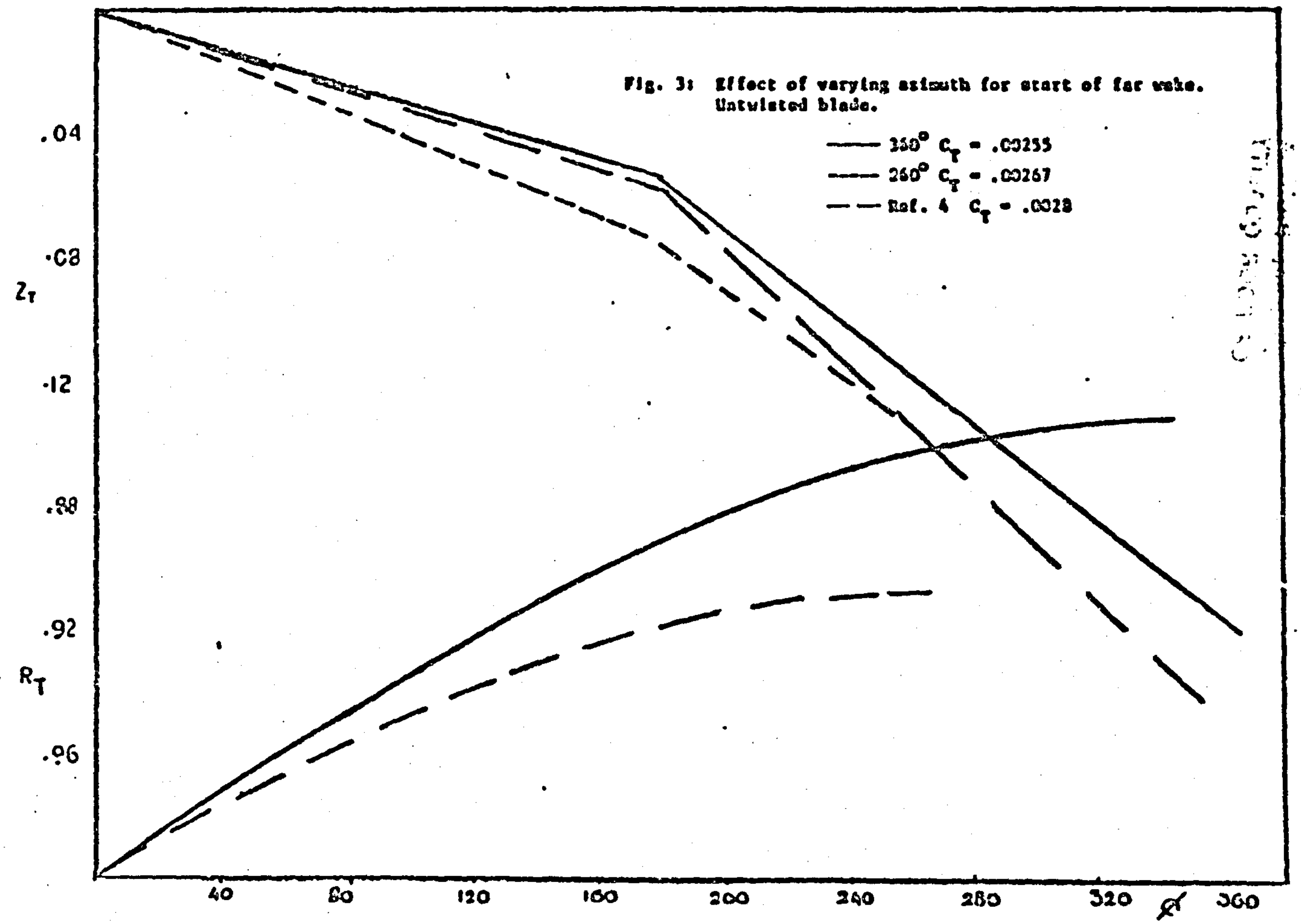


Fig. 3: Effect of varying azimuth for start of far wake. Untwisted blade.

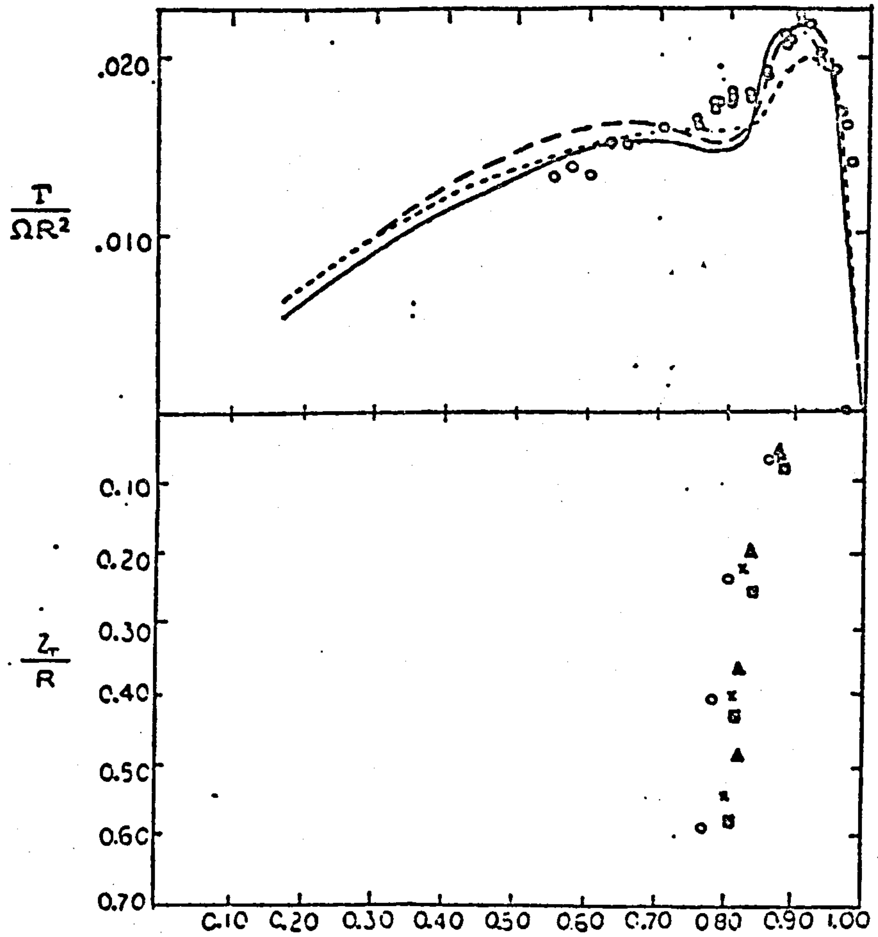


Fig. 4: Effect of the parameter K

Δ ——— K = 0 C_T = 0.00452

X — — K = .5 C_T = 0.00464 O experimental C_T = 0.0460

⊠ ——— K = 1 C_T = 0.00448

$\epsilon_1 = 0.05$, $\epsilon_2 = 0.01$

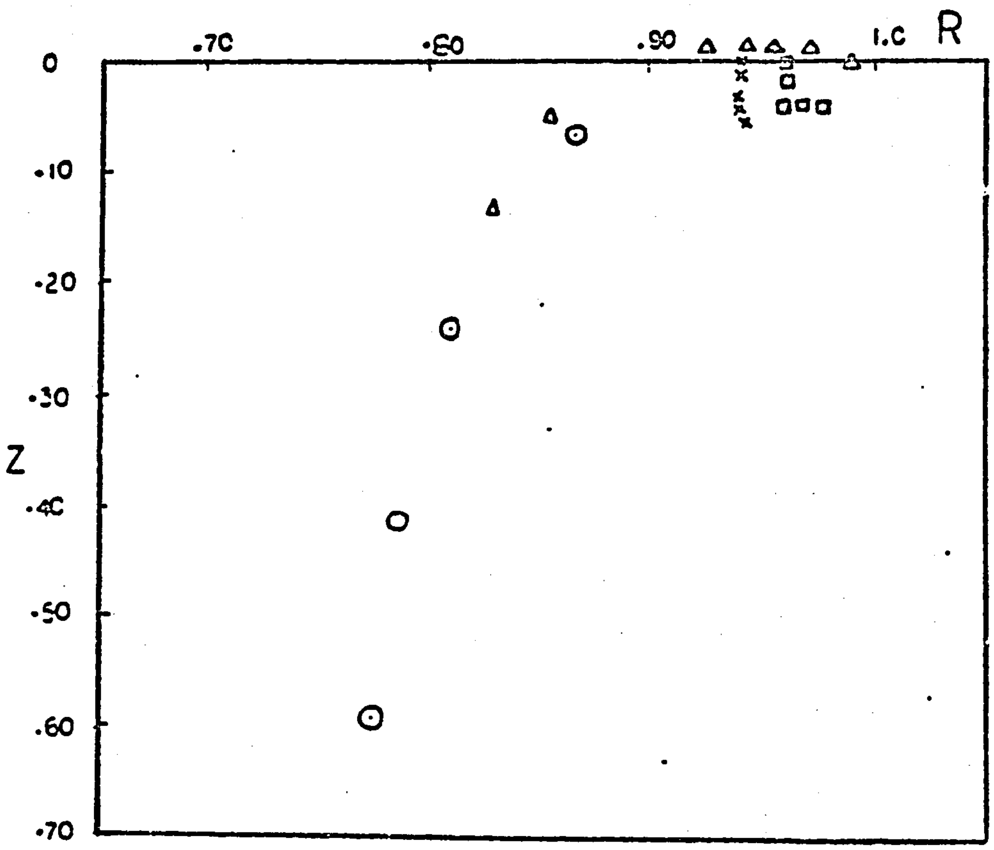


Fig. 5: Behavior of vortices near the tip of the blade showing start of roll up

Δ —— vortex at η = .985

□ — — vortex at η = .95

x —— vortex at η = .93

⊙ —— test vortex at η = ,985

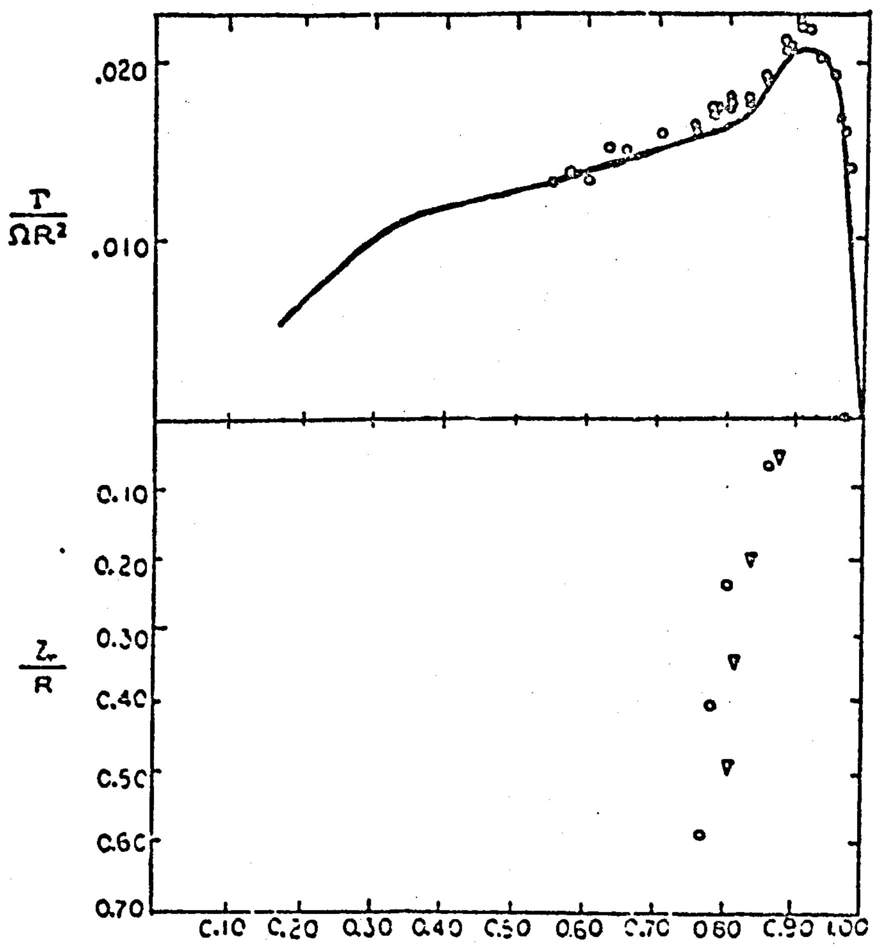


Fig. 6: Comparison of experimental and theoretical bound circulation distribution and geometry. Near wake extends to 40° azimuth and integrated.

$C_T = 0.00448$, 0 test $C_T = 0.0046$, ∇ computed geometry

$c_1 = 0.05$, $c_2 = 0.01$

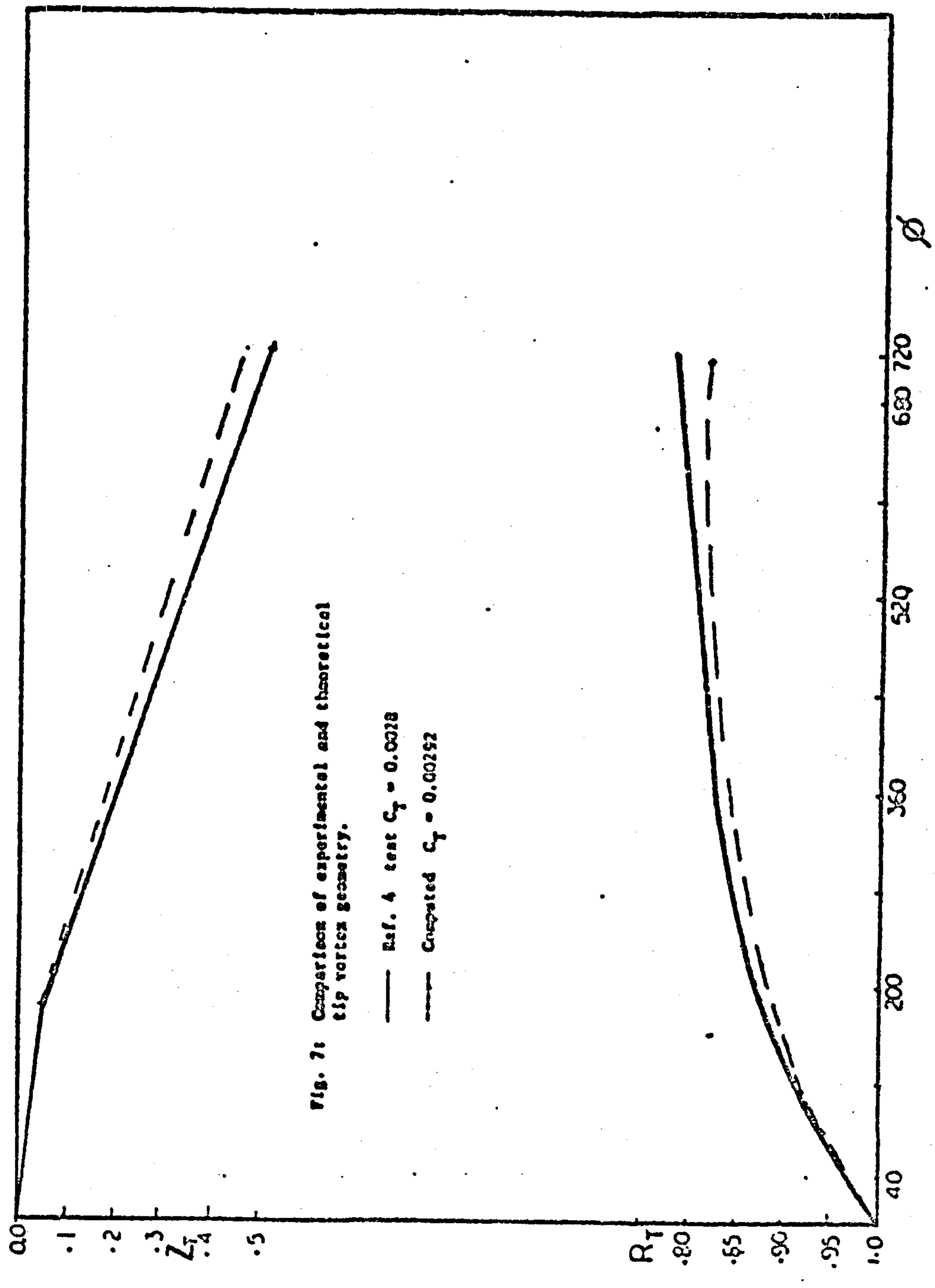


Fig. 7: Comparison of experimental and theoretical tip vortex geometry.

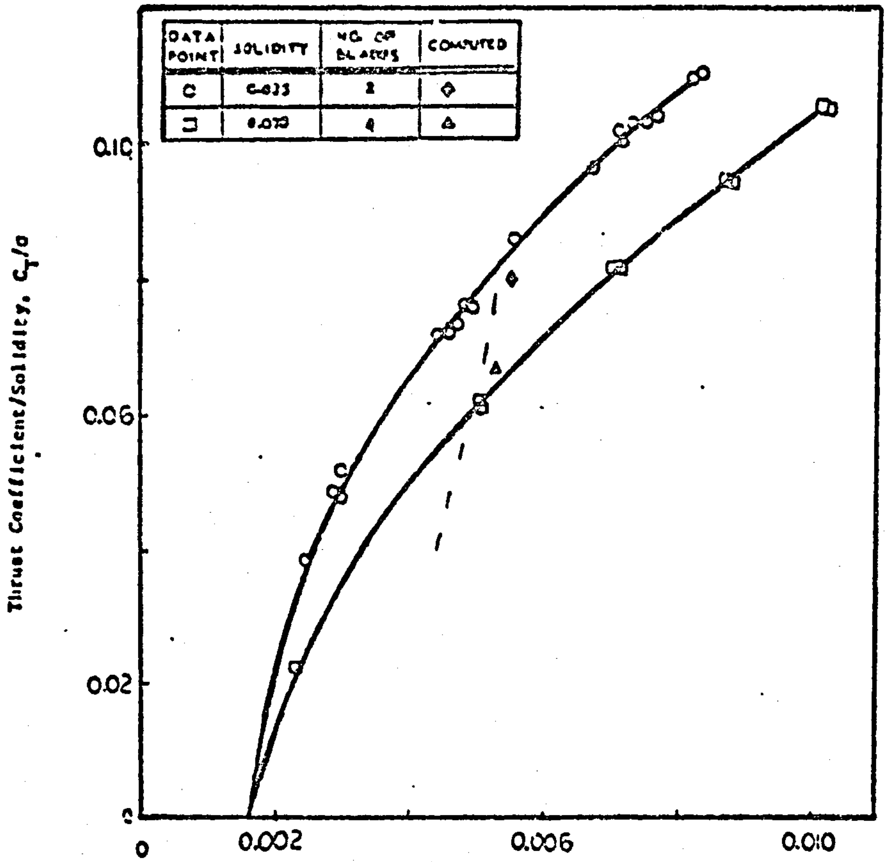


Fig. 8: Torque coefficient/solidity, C_Q/σ

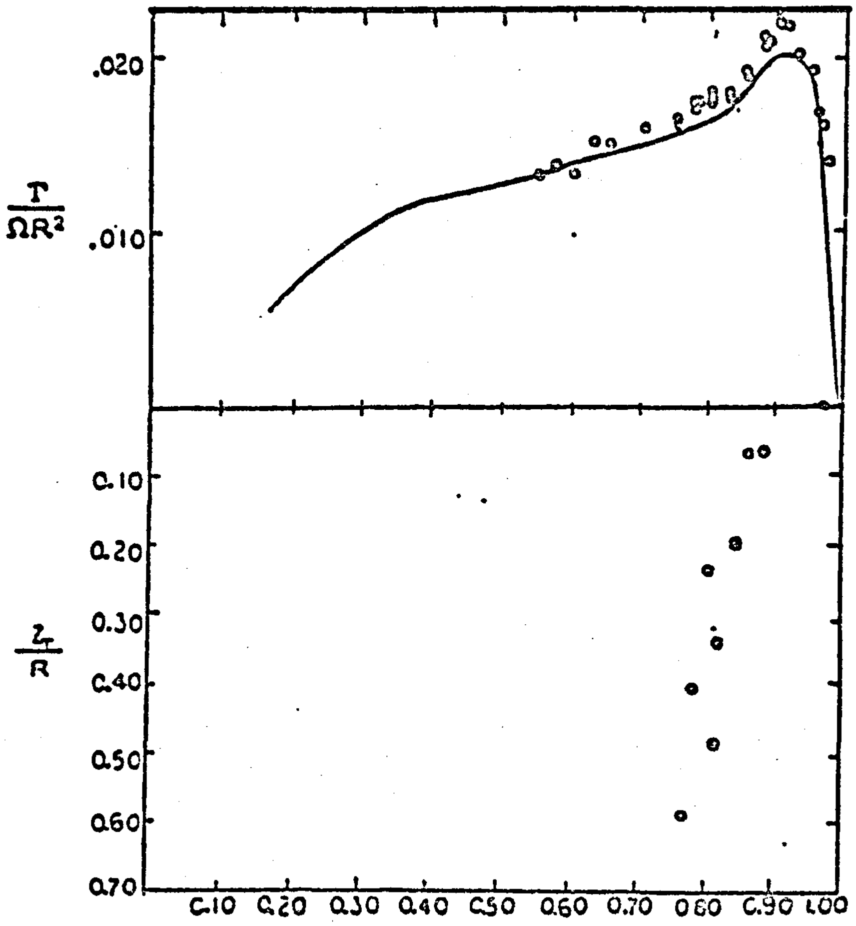


Fig. 9: Comparison of test and theoretical bound circulation distribution and wake geometry. Near wake extends to 40° azimuth and fixed on the plane of rotation.

C_T = 0.00445

● computed geometry

○ test data C_T = 0.0046

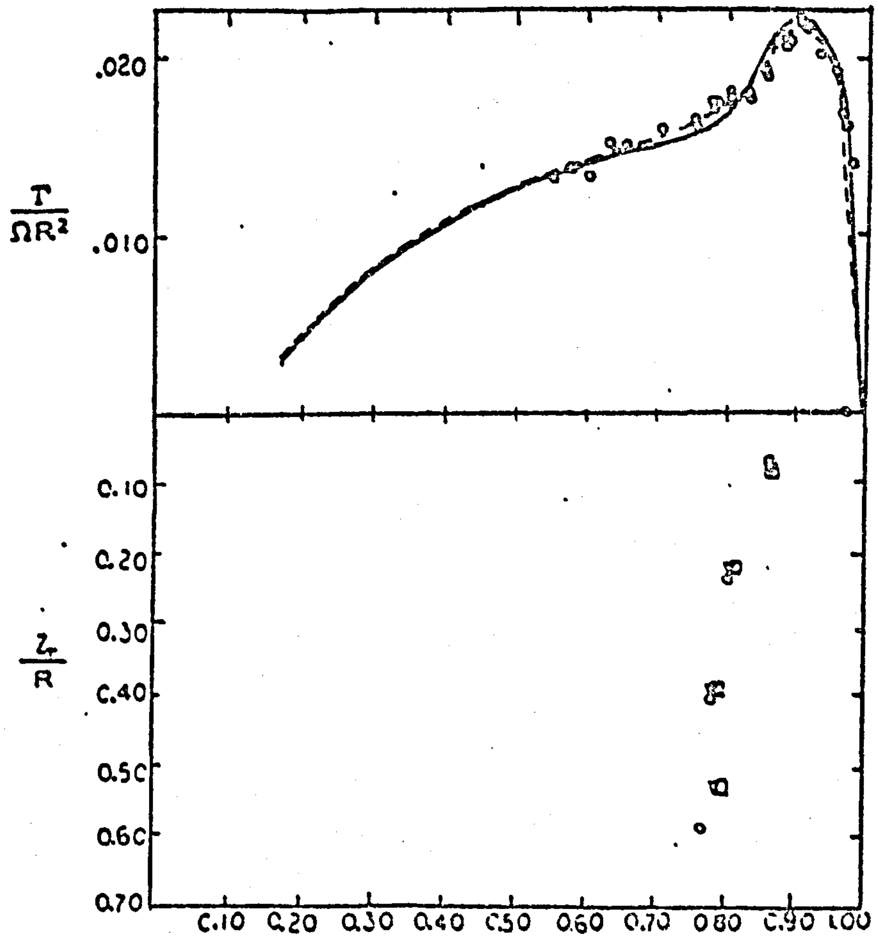


Fig. 10: Effect of varying the azimuthal size of near wake in the fourth vortex sheet model.

□——— 40° C_T = 0.00456

▽——— 30° C_T = 0.00453

experimental C_T = 0.0046

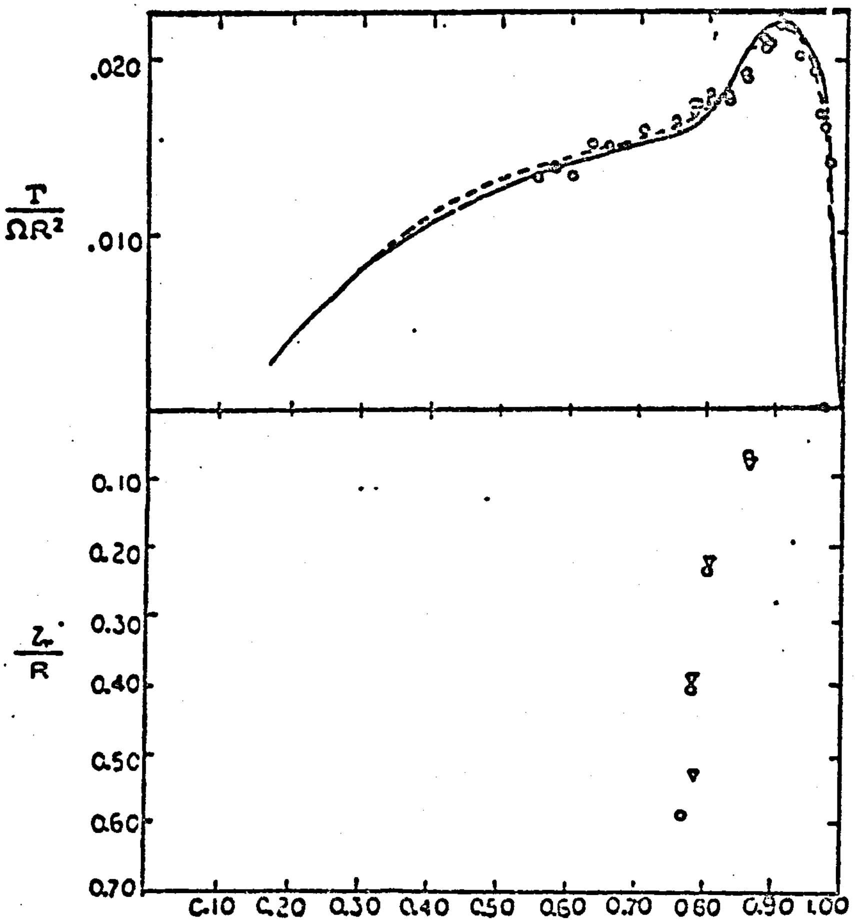


Fig. 11: Effect of different roll up schedule of 1st middle rolled up vortex on the bound circulation and wake geometry.

---- from peak to 82.3% $C_T = 0.00459$

—— from peak to 71.5% $C_T = 0.00453$

▽ computed geometry (close for both cases)

experimental $C_T = 0.0046$

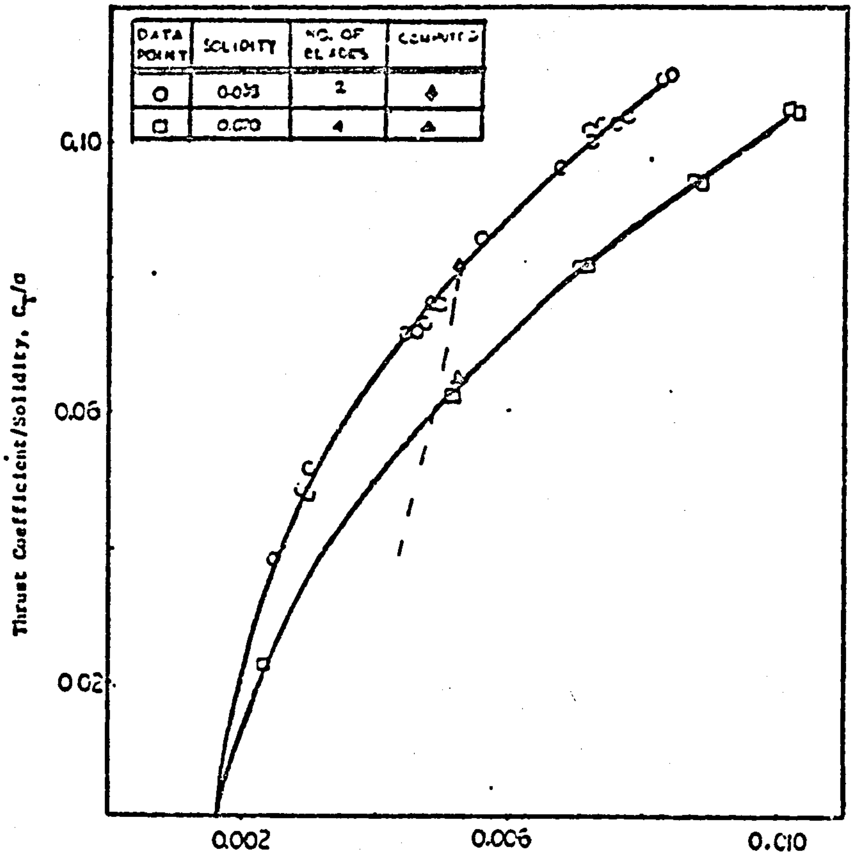


Fig. 12: Torque coefficient/solidity, C_Q/σ

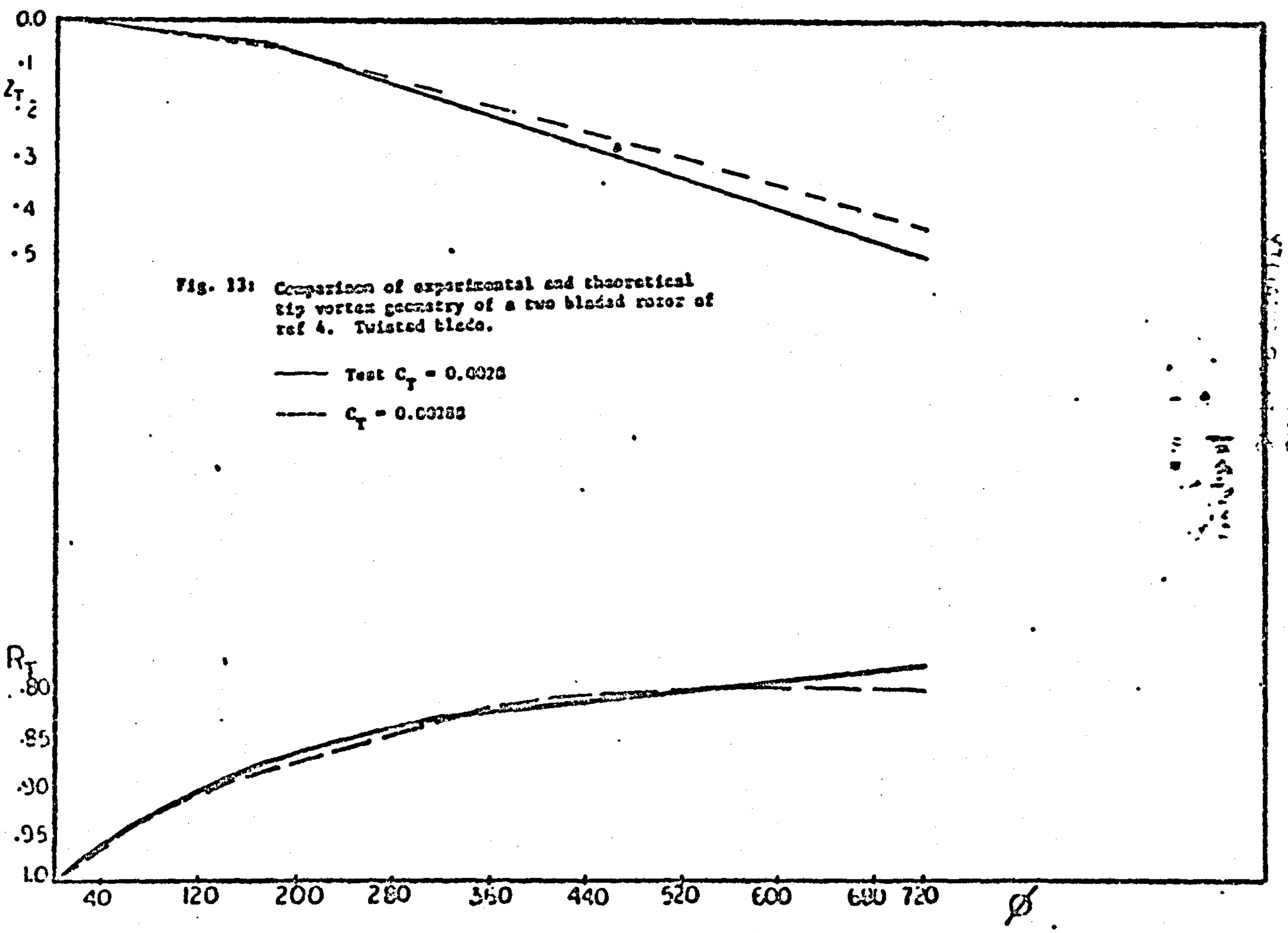


Fig. 13: Comparison of experimental and theoretical tip vortex geometry of a two bladed rotor of ref 4. Twisted blade.

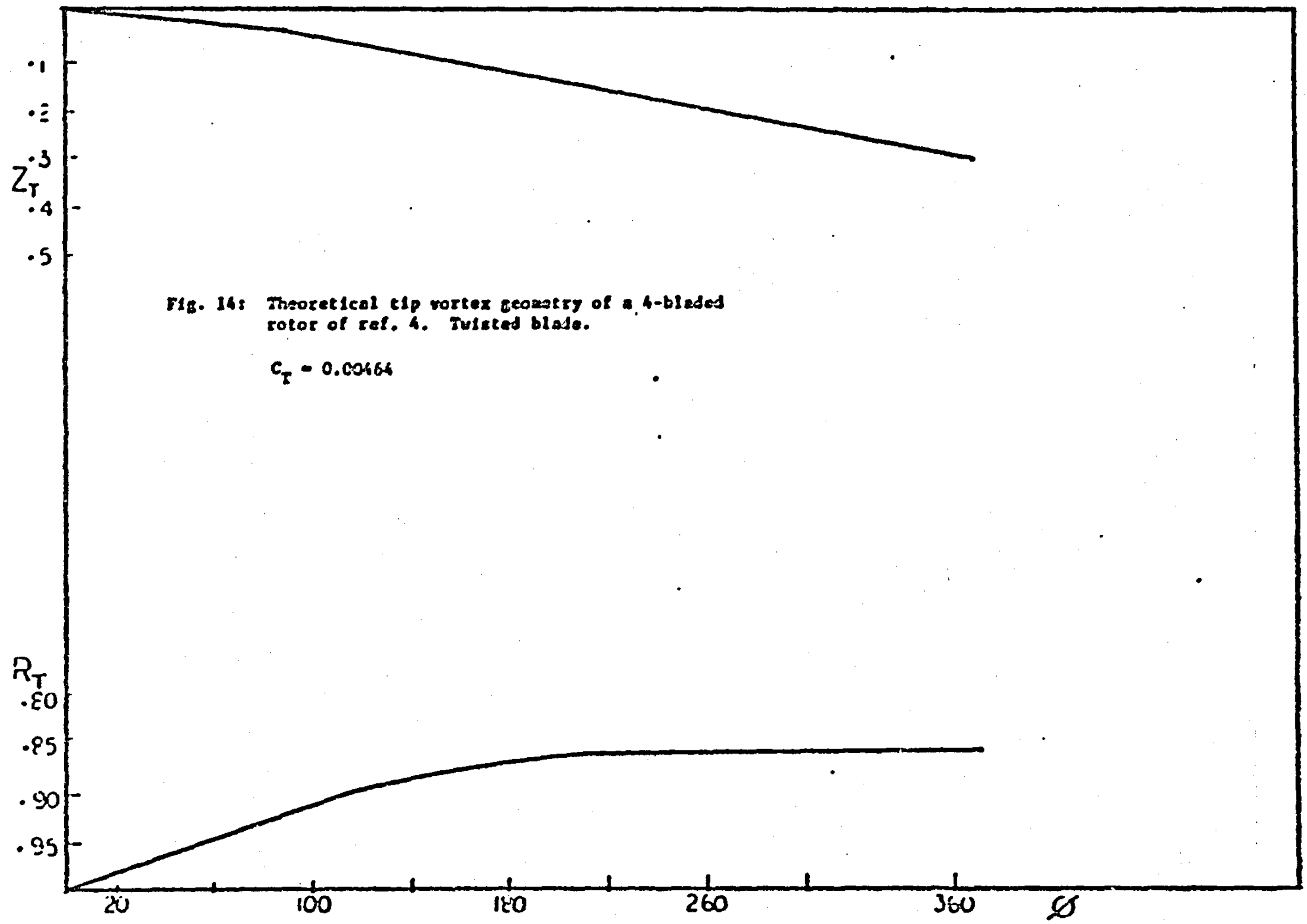


Fig. 14: Theoretical tip vortex geometry of a 4-bladed rotor of ref. 4. Twisted blade.

$C_T = 0.00464$

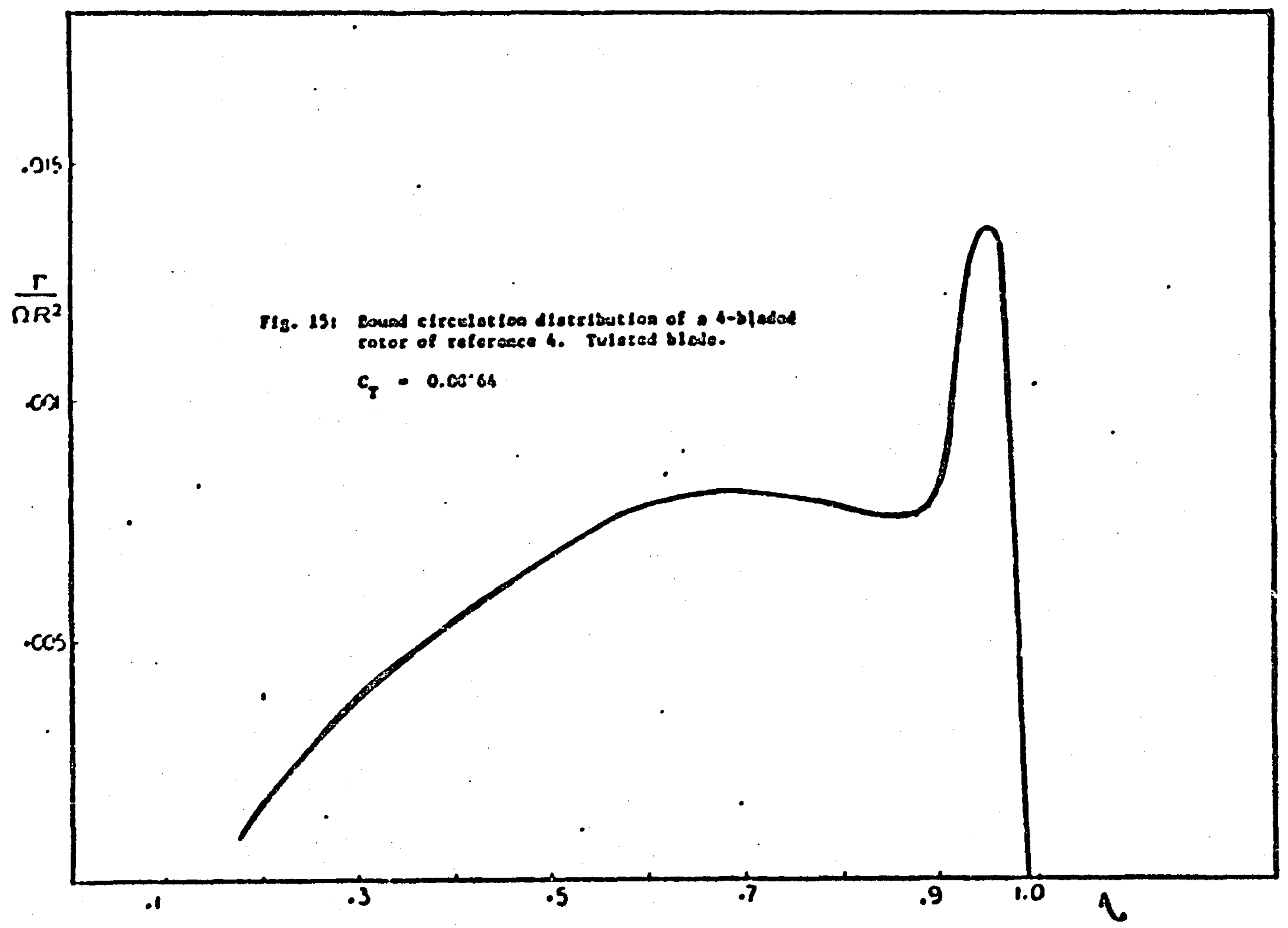


Fig. 15: Bound circulation distribution of a 4-bladed rotor of reference 4. Twisted blade.

C_T = 0.0064

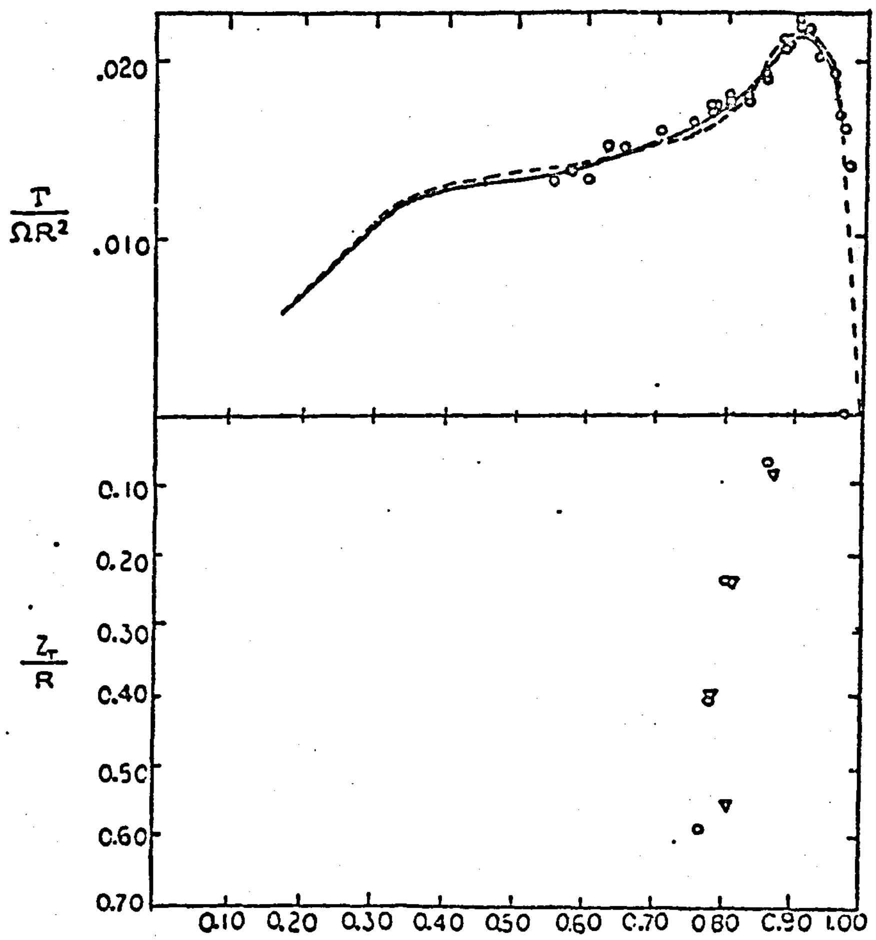


Fig. 16: Effect of the slope lift curve

- - - - 2π C_T = 0.00471 o experimental C_T = 0.0046

——— 6.15 C_T = 0.00465 ∇ computed geometry (close for both cases)

$\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.015$

APPENDIX I

ORIGINAL PAGE IS
OF POOR QUALITY

Self-Element Induced Velocity Computation

The Biot-Savart Law shows that as the point where the determination of induced velocity is required approaches its corresponding line vortex, a singularity is expected. Consequently, a special technique is developed to calculate the vertical induced velocity at the two end points (or nodes) of a curved vortex segment. A basic assumption of this approach is to consider the helical vortex filament as a vortex ring. From reference 6, the vertical component of velocity induced at η due to a ring of radius r is

$$W = \frac{\Gamma}{4\pi R} \int_0^{2\pi} \frac{r(r - \eta \cos \phi)\, d\phi}{(\eta^2 + r^2 - 2r\eta \cos \phi)^{3/2}} \tag{1}$$

The limit of integration has been changed to α and $2(\pi - \alpha)$ where the value of α is half the size of the vortex segment. This will avoid the singularity that occurs with ϕ equal to zero and r equal to η. The integral interval is the same as the size of a vortex segment. After integrating this equation numerically, the result is the sum of vertical induced velocity due to that portion of the ring from α to $2(\pi - \alpha)$ The self-induced velocity is then obtained by subtracting the above result from the vortex ring solution given in reference 8, which derives a self-induced velocity on the vortex ring expressed as

$$W_s = \frac{\Gamma}{4\pi R}\left(\ell n(8r/\varepsilon) - 1/4\right) \tag{2}$$

where ϵ is the core radius. A core radius of approximately $\epsilon/r = .02$ has been used as in reference 6, giving

$$W_s \approx \frac{\Gamma}{4\pi R}(6) \qquad (3)$$

APPENDIX II

The computer code is written in Fortran language. The program takes between 10 and 15 minutes to run on a Honeywell 68/dps for the third and fourth vortex sheet models. Listing of the subroutines and the main program are found in Table 1. Table 2 defines the variables in alphabetical order and the output variables are listed in Table 3. A typical solution is shown in Table 4 and the overall flow chart is given in Table 5. An input file is required to run the program and Table 6 gives a typical input data file for the case in figure 9.

Note that in the main program, the number of rolled up vortices and the type of vortex sheet model must be specified (the values of nrl2, nrl1, irl1, irl2 and imodel are required). The roll up process is coded in the subroutines loop22, loop0 and compa2. The convergent test is 3% for the bound circulation and 6% for the wake geometry.

TABLE 1

Main Program (fwmain2)

The main program doesn't carry out any calculation. It controls the execution of the program by calling the subroutines. It also defines the parameters such as what type of vortex sheet model and how many rolled up vortices are in the case.

List of Subroutines

input2: This is concerned with the input of the case and verifies the input data. It also computes the constants used by the other subroutines.

semri2: The semi-rigid wake of rotor hovering aerodynamics is analyzed. The results are used to initialize the free wake analysis.

compa2: The function is to modify the results of the semi-rigid wake to include the initial roll up process.

inivnd2: Computes the coefficients which are required to determine the strength of the vortex elements (due to unit circulation distribution)

COFN2: Computes the coordinates of the centers in the near and intermediate wakes using the coordinates of the nodes loop0: Determines the induces velocity at the blade centers due to the intermediate and far wake vortices

loop12: Computes the induced velocity at the blade centers from the continuous vortex sheet in the near wake. It also loops on the bound circulation and tests the convergence

loop22: Computes the strength of the vortex elements and also controls the computation of induced velocities everywhere in the wake

vind2: loop on the nodes where the induced velocities are computed. The entries are named as follows:

<u>Entry vindb2:</u>

Induced velocities on the blade are determined.

<u>Entry vindn2:</u>

Induced velocities on the near wake centers are determined

<u>Entry vindnl2:</u>

Induced velocities on the near wake rolled up nodes are computed

<u>Entry vindl2:</u>

Induced velocities on the intermediate wake rolled up nodes are determined indvel: Calculates the induced velocities due to a rectangular segment or a semi-infinite cylinder element, depending on the value of "it" variables. It has 2 entries: wxyz and coord.

prtv: Initialize the induced velocity everywhere in the wake. The entry used is clv2.

wnofc2: Interpolates the induced velocities at the centers to the nodes in the continuous near wake vortices. Computes the total induced velocities at the anticipated rolled up near wake vortices and at the rolled up intermediate wake vortices. Also transforms the x,y,z components of velocities into the radial, tangential and axial components. This subroutine is not used for the 2nd vortex sheet model

intgr2: Determination of the new vortex element geometry by integrating the induced velocities along the stream lines. This subroutine is not used for the 2nd vortex sheet model

wnofc3: Same as in wnofc2, but with the addition of continuous near wake vortices. (This is used for 2nd vortex sheet model only.)

intgr3: Same as in intgr2, but with the addition of continuous near wake vortices. (This is used for 2nd vortex sheet model only.)

out2: Computes the performance of the helicopter rotor and prints the output of the results

input data: Input file of parameters needed to run the program

```
c    *  ****************
c    *  *         .    *
c    *  *  PROGRAM FWCMAIN2  *
c    *  *              *
c    *  ****************
c    *
c    *  FREE-WAKE ANALYSIS OF ROTOR HOVERING AERODYNAMICS
c    *  ------------------------------------------------
c    *
c    *
c    *       12/1/80
c    *
c    *  SUBROUTINES :
c    *
c    *  INPUT2(PUNCH)
c    *  SEMRI2
c    *  COMPA2
c    *  INIVND2
c    *  COFN2
c    *  LOOP22
c    *  VIND2(VINDB2,VINDN2,VINDN12,VINDI2,EXECH,INICH)
c    *  INDVEL(COORD,WXYZ)
c    *  LOOP12
c    *  PRTV(PRTCEN,PRTNOD,PRTVD,CLV,PRTVN,PRTVI)
c    *  INTGR2
c    *  WNOFC2(INIWN2)
c    *  WNOFC3(INIWN3)
c    *  INTGR3
c    *  OUT2(OUTINT2)
c    *
c    *
c    *************************************
c    *
        common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
         &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamat,
         &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
         &ipr,iplott
        common/hparm/ifmu
        common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
         &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
         &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
         &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
         &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
        &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel
        common /farc/eta(40),wx(22,40),wy(22,40),wz(22,40)
        equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
        equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
        namelist/nparm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,
         &itrcl2,itrcg,itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,
         &itrans,ipn,jpunch,iplotw,isw,icont,iplotg,ipr,iplott,ifmu,
         &lims,lim1,lim2,isem
c    *
c    *************************************
c    *
c    *  DEFAULT PARAMETERS
c    *
        nrl1=25
        nrl2=27
        irl1=2
        irl2=4
```

```
            imodel=3
            iwr=6
            ird=5
            ipn=9
            ipr=6
            itrace=0
            ipscm=1
            itrcl1=0
            itrcl2=0
            itrcg=0
            itrcnt=0
            itrcf=0
            itrtg=0
            ivergr=0
            itrans=0
            jpunch=0
            icont=0
            icgen=0
            nmes=0
            ifmu=0
            isem=0
            coeff=0.
c    *
c    *  DATA BY DEFAULT
c    *
            nblds1=2
            itwist=0
            alphas=1.
            coeff1=.5
            lims=50
            lim1=50
            lim2=15
c    *
c    *  INPUT OF THE OPTIONS
            read(5,nparm)
            if(isem.eq.1)write(iwr,55)
55          format(" SEMI-RIGID WAKE ANALYSIS ONLY FOR ALL CASES",//)
            jcont=icont
            limd=lim1
            ltrcw=itrcw
            ltrcl2=itrcl2
            lmes=nmes
c    *
c    *
            if(imodel.eq.2) write(6,1003)
            if(imodel.eq.3) write(6,1004)
            if(imodel.eq.4) write(6,1005)
c    *
1003        format(///," SECOND VORTEX SHEET MODEL ",///)
  1004      format(///," THIRD VORTEX SHEET MODEL ",///)
  1005      format(///," FOURTH VORTEX SHEET MODEL ",///)
c    ****************************************************************
c    *
c    *  LOOP ON THE CASES SUMMITED TO THE PROGRAM
c    *  ----------------------------------------
c    *
            do 999 ncas=1,100
c    *
c    *  INPUT OF THE DATA FOR THE CURRENT CASE.
c    *  TERMINATION IF AN END OF FILE IS ENCOUNTERED ON (IRD)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          call input2(ncas)
          liml=limd
          itrcw=itrcw
          itrcl2=itrcl2
          nmes=lmes
c    *
          jter=1
          if(jcont.eq.1) jter=niter+1
c    *
c    *   INITIALISATIONS
          if(isem.eq.1) goto 17
c    *
c    *   INITIALIZATION FOR VIND2
          if(isamet.eq.0) call inivnd2
c    *   INITIALISATION FOR WNOFC
          if(isamet.eq.0.and.imodel.eq.2) call iniwn3
          if(isamet.eq.0.and.imodel.ne.2) call iniwn2
c    *
          if(jcont.ne.0) go to 10
c    *
c    *   SEMI-RIGID WAKE ANALYSIS
17        continue
          call semri2
          if(isem.eq.1) goto 999
c    *   IF ITEST=1 CONVERGENCE IS NOT REACHED IN SEMRI,
c    *   THE CASE IS TERMINATED
          if(itest.eq.1) goto 999
c    *
c    *   DATA INITIALISATION FROM THE SEMI-RIGID WAKE ANALYSIS
          call compa2
          call cofn2
          if(itrace.eq.1) write(iwr,102)
102       format("*** MAIN *** NODES GENERATED BY COMPA")
          if(itrace.eq.1) call prtnod2
c    *
10        continue
          if(icont.ne.3) goto 13
          call compa2
          call cofn2
13        continue
          jcont=0
c    *
c    *
c    *   MAIN LOOP
c        MAP OF THE FAR WAKE
c
          itest=1
c    *   ---------
c    *
          do 1 niter=jter,lim2
c    *
c    *   VELOCITIES INDUCED AT THE CENTERS AND NODES
          call loop22
c    *
c    *   VELOCITIES AT THE NODES FROM THE VELOCITIES AT THE CENTERS AND TOTAL VELOCITY
          if(itrace.eq.1.and.itrcl2.eq.0) write(iwr,106)
106       format("1*** MAIN ***: VELOCITIES AT THE CENTERS")
          if(itrace.eq.1.and.itrcl2.eq.0) call prtv
c    *
          if(imodel.eq.2) go to 1000
```

```
          call wnofc2
          go to 1001
 1000     call wnofc3
c  *  INTEGRATION (NEW POSITIONS OF THE NODES)
          call intgr3
c  *
c  *  NEW POSITIONS OF THE CENTERS
          call cofn2
          go to 1002
c   *  INTEGRATION (NEW POSITIONS OF THE NODES)
1001      call intgr2
c   *
c   *
c   *  NEW DISTRIBUTION OF CIRCULATION
1002      call lcop12
c   *
c   *  OUTPUT FOR CONTINUATION
          if(jpunch.eq.1) call punch
c   *
c   *  INTERMEDIATE OUTPUT
          if(icgon.eq.1) call outint2
          if(icgon.eq.2) write(iwr,107)
107       format(1h1)
          if(icgon.eq.2) call prtnod2
c   *
c   *  NO CONVERGENCE IN LOOP1, THE CASE IS TERMINATED
          if(itest.eq.2) goto 999
c   *
c   *  CONVERGENCE
          if(itest.eq.0) goto 2
c   *
1         continue
c   *
c   *  NO CONVERGENCE FOR THE MAIN LOOP
          write(iwr,104)lim2
104       format("1*** MAIN ***: NO CONVERGENCE WITHIN",I3," ITERATIONS")
          call out2(1)
          goto 3
c   *
c   *  OUTPUT OF THE FINAL RESULTS
c   *  --------------------------
c   *
2         continue
          iout=2
          write(iwr,105) nitor
105       format("1*** MAIN ***: FINAL RESULTS - CONVERGED AFTER",I4,
         &" ITERATIONS")
          do 70 lcd=1,4
          cdk=0.5*lcd
          if(lcd.eq.4) iout=1
          call out2(iout)
70        continue
3         continue
999       continue
c   *
          stop
          end
```

```
      subroutine input2(ncas)
c  *
c  *  *********************
c  *  *                   *
c  *  *  SUBROUTINE INPUT2 *
c  *  *                   *
c  *  *   PROGRAM FWC     *
c  *  *                   *
c  *  *********************
c  *
c  *
c  *  12/1/80
c  *  INPUT OF THE DATA FOR THE CURRENT CASE.THEN COMPUTES
c  *  THE CONSTANTS USED BY MORE THAN ONE SUBROUTINE
c  *  EXIT WHEN AN END OF FILE IS ENCOUNTERED ON (IRD)
c  *
c  *
c  *  ENTRY PUNCH: THE CURRENT DATA IS WRITTEN ON (IPN)
c  *
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamot,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/hparm/ifmu
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      namelist/case/nblds1,itwist,sigma,knnvr,nnva,dpsind,etan,knivr,
     &niva,dpsiid,etai,thet0d,fmu,thetad,alphas,cd0,cdk,
     &fps1,fps2,coeff1,ntva,iplot,iplotv,itrans,
     &jpunch,iplotw,isw,iplotg,iplott,ifmu
      data conv,twopi/.017453293,6.283185308/
c  *
      if(icont.eq.1) goto 40
      read(ird,case,err=999,end=888)
      nnva=ntva
      dpsin=dpsind*conv
      dpsii=dpsiid*conv
      if(itwist.eq.1) goto 5
      theta0=thet0d*conv
      do 10 i=1,knnvr
      theta(i)=theta0
10    continue
      goto 9
5     do 11 i=1,knnvr
      theta(i)=thetad(i)*conv
11    continue
9     continue
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
          isameb=1
          if(ncas.ne.1.and.nblds.eq.nblds1) goto 77
          isameb=0
          nblds=nblds1
          blades=float(nblds)
          c=cos(twopi/blades)
          s=sin(twopi/blades)
   77     continue
   c   *
          isamat=1
          if(coeff.ne.coeff1) goto 61
          do 62 i=1,nnvr
          if(etan(i).ne.etanv(i)) goto 61
   62     continue
          do 63 i=1,nivr
          if(etai(i).ne.etaiv(i)) goto 61
   63     continue
          goto 64
   61     isamat=0
          coeff=coeff1
          nnvr=knnvr
          nivr=knivr
          nncr=nnvr+1
          nica=niva+1
          nicr=nivr+1
          nival=niva-1
          nnvr1=nnvr-1
          nivr1=nivr-1
          do 65 i=1,nnvr
          etanv(i)=etan(i)
   65     continue
          do 66 i=1,nivr
          etaiv(i)=etai(i)
   66     continue
          do 12 i=2,nnvr
          etanc(i)=(etanv(i)+etanv(i-1))*.5
   12     continue
          do 13 i=2,nivr
          etaic(i)=(etaiv(i)+etaiv(i-1))*.5
   13     continue
          etanc(1)=1.5*etanv(1)-.5*etanv(2)
          etaic(1)=1.5*etaiv(1)-.5*etaiv(2)
          etanc(nncr)=1.5*etanv(nnvr)-.5*etanv(nnvr-1)
          etaic(nicr)=1.5*etaiv(nivr)-.5*etaiv(nivr-1)
   c   *
   64     continue
          if(itrans.eq.0) goto 68
          nnva=nnva+dpsii*2./dpsin
          if((nnva-ntva)*dpsin.lt.dpsii*1.9) nnva=nnva+1
   68     continue
          nnva1=nnva-1
          nnca=nnva+1
          ntva1=ntva-1
          ntca=ntva+1
          do 32 i=1,nncr2
          thetac(i)=theta(i)+(theta(i+1)-theta(i))*
        & (etanc(i+1)-etanv(i))/(etanv(i+1)-etanv(i))
   32     continue
   c   *
   c   *  VERIFICATION OF THE INPUT
```

```
c   *
c   *  NUMBER OF BLADES: LIMIT OF 8 BLADES (SEMRI)
         if(nblds.lt.1.or.nblds.gt.8) goto 60
c   *  LIMITATIONS BECAUSE OF THE ACTUAL DIMENSIONS OF THE ARRAYS
         if(nnvr.gt.25.or.nnvr.lt.2) goto 60
         if(nivr.gt.6.or.nivr.lt.2) goto 60
         if(nnva.gt.44.or.nnva.lt.2) goto 60
         if(niva.gt.50.or.niva.lt.2) goto 60
c   *  VALUES OF ETA RANGE FROM ETA(ROOT) TO 1.
c   *  AND MUST BE INCREASING
         if(etanv(1).lt.0.) goto 60
         if(etanv(1).ne.etaiv(1)) goto 60
         if(etanv(nnvr).ne.1..or.etaiv(nivr).ne.1.) goto 60
         do 80 i=2,nnvr
         if(etanv(i).le.etanv(i-1)) goto 60
80       continue
         do 81 i=2,nivr
         if(etaiv(i).le.etaiv(i-1)) goto 60
81       continue
c   *  ADVANCE RATIO MUST BE POSITIVE
         if(fmu.lt.0.) go to 60
         if(icont.ne.3) go to 255
         read(ird,443) (wxc(i),i=1,nncr2),(wyc(i),i=1,nncr2),
          &(wzc(i),i=1,nncr2),(gammc(i),i=2,nnvr)
         gammc(1)=0.
         gammc(nncr)=0.
443      format(8f10.6)
         write(iwr,443)  (wxc(i),i=1,nncr2),(wyc(i),i=1,nncr2),
          &(wzc(i),i=1,nncr2),(gammc(i),i=2,nnvr)
255      continue
         write(iwr,150)ncas,nblds1,sigma,fmu
150      format("1*** INPUT *** CASE # :",i3,/,
          &" NUMBER OF BLADES",i2,/," SIGMA",f12.5,/,
          &" MU",f12.5)
         if(ltwist.eq.1) write(iwr,151)
         if(ltwist.eq.0)write(iwr,152)
151      format("  BLADES ARE TWISTED")
152      format("  BLADES ARE NOT TWISTED")
         if(ltwist.eq.0) write(iwr,161) thet0d
161      format(" PITCH ANGLE=",f10.5," DEGREES")
         if(ltwist.eq.1) write(iwr,162) (thetad(i),i=1,knnvr)
162      format(" PITCH ANGLE DISTRIBUTION(DG):",5f10.5,/,5x,6f10.5,
         &/,5x,4f10.5)
         write(iwr,153)alphas,cd0,cdk
153      format(" STALL ANGLE",f12.5,/,"  CD=",f5.4,"+",f5.3,"*ALPHA**2")
         write(iwr,154)coeff1,lims,lim1,lim2,fps1,fps2
154      format(" ROOT AND TIP VORTICES FACTOR:",f10.5,
         &/,"&MAX.&NUMBER&OF&ITERATIONS&FOR&SEMRI,LOOP1,LOOP2:",3i3,/,
          &" CORE SIZE FOR SURFACE ELEMENTS:",f10.5," FOR SEGMENT ELEMENTS:"
          &,f10.5)
         write(iwr,155)knnvr,ntva,(etan(i),i=1,knnvr)
155      format(" NEAR WAKE DEFINITION:(",i2,",",i2,"),",4f10.5,
         &/,5x,6f10.5,/,4f10.5)
         write(iwr,156)knivr,niva,(etai(i),i=1,knivr)
156      format(" INT. WAKE DEFINITION:(",i2,",",i2,"),",4f10.5,
         &/,5x,6f10.5,/,5x,4f10.5)
         write(iwr,170)nnva,ntva
170      format(" NNVA",i3," NTVA",i3)
         write(iwr,180) isameb,isamet
180      format(" ISAMEB",i3," ISAMET ",i3)
```

```
      if((icont.eq.0).or.(icont.eq.3)) return
c  *
      jcont=icont
c  ***********************************************
c  *
40    continue
      rewind ipn
      read(ipn,err=999,end=777) tparm,tgeom,tvind,twnof
      read(ipn,err=999,end=777) tresul
      isamob=1
      isamat=1
      if(jcont.eq.2) read(ird,case,err=999,end=888)
      return
777   write(iwr,191)
191   format(" *** INPUT ***: ERROR,ICONT=1, EOF ON IPN")
      stop
c  *
c  *****************************************
c  *
      entry punch
c  *
      rewind ipn
c  *  NOTE: IBM LIMITATION OF 32K BYTES OF DATA/RECORD
      write(ipn) tparm,tgeom,tvind,twnof
      write(ipn) tresul
c  *  NOTE: THIS REWIND IS NECESSARY, WITHOUT IT THE LAST RECORD
c  *      IS NOT WRITTEN IN CASE OF ABEND
      rewind ipn
      if(itrace.ge.1) write(iwr,190)
190   format(" *** PUNCH ***")
      return
c  *
c  ******************************************
c  *
888   write(iwr,201)
201   format(1h1)
      stop
999   write(iwr,200)
200   format(" *** INPUT *** ERROR ON (IRD) ")
c  *
99    format(2i2,e10.5)
102   format(8f10.5)
122   format(3i2)
      stop
c  *
60    continue
      write(iwr,181)
181   format(" *** INPUT *** DATA INVALID OR OUT OF RANGE ")
      write(iwr,case)
      stop
c  *
      end
```

```
      subroutine semri2
c   *
c   *  ************************
c   *  *                      *
c   *  *    PROGRAM FWC      *
c   *  *                      *
c   *  *   SUBROUTINE SEMRI2   *
c   *  *                      *
c   *  ************************
c   *
c   *       12/1/80
c   *
c   *  THIS SUBROUTINE USES THE SEMI-RIGID MODEL
c   *  TO COMPUTE A FIRST APPROXIMATION OF THE GEOMETRY OF THE WAKE
c   *  AND OF THE DISTRIBUTION OF CIRCULATION ALONG THE BLADES
c   *
c   *  THE SUBROUTINE RETURNS ITEST1=0
c   *  IF CONVERGENCE IS REACHED IN (LIMS) ITERATIONS
c   *  ITEST1=1 IF NO CONVERGENCE
c   *
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/hparm/ifmu
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,29),wync(26,29),wznc(26,29),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      dimension alpha(24),wyct(24),wzct(24),wyv(25),
     &wzv(25),sn(36,8),cn(36,8)
      dimension flamda(24),u(24)
      reallip(24),lp,lt,mu,tlip(24),flip(24),dp(24),fdp(24),tdp(24),
     &tp(24),fp(24)
      dimension bigf(25),fmp(25)
      dimension sav1(24),sav2(24),facv(24)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      data phi5,phi10,pi,fpi,twopi,pi2/.087266463,.174532929,
     &3.141592653,.07957747151,6.283185308,9.869604404/
      data lwx,lwy,lwz /"WX  ","WY  ","WZ  "/
      data leta,lu,llamda/"ETA ","U   ","LAM "/
      data lthet,lalpha,lgamma/"THET","ALPH","GAMM"/
      data llip,ltlip,lflip/"LIP ","TLIP","FLIP"/
      data ldp,ltdp,lfdp/"DP  ","TDP ","FDP "/
      data ltp,lfp/"TP  ","FP  "/
      data lbf/"F   "/
      data lfac/"FACV"/
c   *
c   *
c   *  ARRAY OF SINES AND COSINES
```

```
c   *
        do 10 nb=1,nblds
        psi=(nb-1)*twopi/blades
        do 10 j=1,36
        phis=phi5-psi+phi10*(j-1)
        sn(j,nb)=sin(phis)
        cn(j,nb)=cos(phis)
10      continue
c   *
c   *  PRANDTL'S CORRECTION FACTORS FOR THE NUMBER OF BLADES
c   *
        fld=fmu-sigma*pi/4.
        do 20 i=1,nncr2
        fmp(i)=.5*(fld+sqrt(fld**2+sigma*pi*etanc(i+1)*thetac(i)))
        fs=blades*.5*sqrt(1.+fmp(i)**2)/fmp(i)
        smallf=fs*(1.-etanc(i+1))
        if(smallf.gt.10.) go to 1000
        sma=exp(-smallf)
        go to 1001
1000    sma=0.
1001    bigf(i)=2.*acos(sma)/pi
20      continue
        if(lpsem.eq.2) write(iwr,110)lbf,(bigf(n),n=1,nncr2)
c
c      INITIAL APPROXIMATION
        itest=0
        fkp=.5*(fld+sqrt(fld**2+sigma*pi*theta0))
        wi=abs(fkp-fmu)
        do 31 i=1,nncr2
        wyc(i)=0.
        wzc(i)=fmp(i)-fmu
        facv(i)=.8
31      continue
        do 32 i=1,nnvr
        wyv(i)=0.
        wzv(i)=fmp(i)-fmu
32      continue
c   *
c   *  SPANWISE MOMENTUM THEORY
c   *
        ca=.25
        do 16 k=1,10
        do 15 i=1,nncr2
        if(k.eq.1) goto 650
        f1=atan((wzc(i)+fmu)/(etanc(i+1)+wyc(i)))
        wyc(i)=wyc(i)*f1/flamda(i)
650     continue
        tglam=(wzc(i)+fmu)/(etanc(i+1)+wyc(i))
        flamda(i)=atan(tglam)
        alphal=flamda(i)-thetac(i)
        if(alphal.gt.alphas)alphal=alphas
        if(alphal.lt.-alphas) alphal=-alphas
        wzct(i)=-(pi*.25)*sigma*
         &sqrt((etanc(i+1)+wyc(i))**2+(fmu+wzc(i))**2)*
         &alphal/(tglam*etanc(i+1)) /bigf(i)
        wzc(i)=wzct(i)*.25+wzc(i)*.75
        wzc(i)=(wzct(i)-ca*wzc(i))/(1.-ca)
        wyc(i)=-wzc(i)*tglam
15      continue
16      continue
```

```
          if(ipsem.ne.2) goto 18
          write(iwr,110)iwz,(wzct(n),n=1,nncr2)
          write(iwr,110)ilamda,(flamda(n),n=1,nncr2)
          write(iwr,110)iwy,(wyc(n),n=1,nncr2)
          write(iwr,110)iwz,(wzc(n),n=1,nncr2)
18        continue
c    *
c    *
          do 17 n=2,nnvr1
      aaa=(etanv(n)-etanc(n))/(etanc(n+1)-etanc(n))
          wzv(n)=wzc(n-1)+(wzc(n)-wzc(n-1))*aaa
          wyv(n)=wyc(n-1)+(wyc(n)-wyc(n-1))*aaa
17        continue
          wyv(1)=wyc(1)
          wyv(nnvr)=wyc(nncr2)
          wzv(1)=wzc(1)*.5
          wzv(nnvr)=wzc(nncr2)*.5
c    *
c    *  MAIN LOOP
c    *
          gammc(1)=0.
          gammc(nncr)=0.
          do 400 niter=1,lims
c    *
          e=.10*wi
          eps=.0005*wi
          gtmin=0.
          gtmax=0.
c    *
c    *  CALCULATION OF THE ANGLES AND CIRCULATIONS
c    *
          do 40 i=1,nncr2
          wxc(i)=0.
          wyct(i)=0.
          wzct(i)=0.
c    *
c    *  CORRECTION TO ACCELERATE THE CONVERGENCE
          fl=atan((wzc(i)+fmu)/(etanc(i+1)+wyc(i)))
          ccr=fl/flamda(i)
          if(ccr.gt.1.1) ccr=1.1
          if(ccr.lt..9) ccr=.9
          wyc(i)=wyc(i)*ccr
c    *
c    *  INFLOW ANGLE
          flamda(i)=atan((wzc(i)+fmu)/(etanc(i+1)+wyc(i)))
          u(i)=(etanc(i+1)+wyc(i))/cos(flamda(i))
c    *
c    *  ATTACK ANGLE
          alpha(i)=flamda(i)-thetac(i)
c    *  STALL EFFECT
          alphal=alpha(i)
          if(alpha(i).le.-alphas) alphal=-alphas
          if(alpha(i).gt.alphas) alphal=alphas
c    *
c    *  CIRCULATION ALONG THE BLADE
          gammc(i+1)=pi*pi*sigma*u(i)*alphal/blades
c    *
c    *  STRENGH, FOR THE FAR WAKE
          if(gtmin.ge.gammc(i+1))gtmin=gammc(i+1)
          if(gtmax.lt.gammc(i+1))gtmax=gammc(i+1)
```

```
40        continue
          gtt=-gtmax
          if(-gtmin.gt.gtmax)gtt=-gtmin
c   *
c   *
c   *   CALCULATION OF THE VELOCITIES INDUCED BY THE NEAR WAKE
c   *
          do 50 mm=1,nnvr
          tglam=(wzv(mm)+fmu)/(etanv(mm)+wyv(mm))
          wm2t=wzv(mm)+fmu
          etav=etanv(mm)
          da=phi10*etav
          gammat=gammc(mm+1)-gammc(mm)
          ff=da*gammat*fpi
          do 50 nn=1,nncr2
          etap=etanc(nn+1)
          wy2=0.
          wx2=0.
          wz2=0.
          do 51 nb=1,nblds
          do 51 jj=1,9
          sp=sn(jj,nb)
          cp=cn(jj,nb)
          f=etap-etav*cp
          h=wm2t*(phi10*jj-phi5)
          r0=sqrt(f*f+(etav*sp)**2+h*h)
          the=da*.5/r0
          aa=the*the*(tglam*tglam+1.)
          bb=the*(h*tglam+etap*sp)/r0
          fj=.5*((aa+bb)/sqrt(1.+2.*bb+aa)+(aa-bb)/sqrt(1.-2*bb+aa))/
           &(aa-bb*bb)
          fu=ff*fj/(r0*r0*r0)
          wy2=wy2+fu*(tglam*f-h*sp)
          wx2=wx2+fu*(etav*sp*tglam-h*cp)
          wz2=wz2+fu*(etav-etap*cp)
51        continue
          wxc(nn)=wxc(nn)+wx2
          wyct(nn)=wyct(nn)+wy2
          wzct(nn)=wzct(nn)+wz2
50        continue
c   *
c   *   CALCULATION OF THE VELOCITIES INDUCED BY THE FAR WAKE
c   *
          do 60 mm=1,nnvr,nnvr1
          tglam=(wzv(mm)+fmu)/(etanv(mm)+wyv(mm))
          wm2t=wzv(mm)+fmu
          etav=etanv(mm)
          da=phi10*etav
          ff=gtt*da*fpi
          if(mm.eq.1) ff=-ff
          do 60 nn=1,nncr2
          etap=etanc(nn+1)
          wx2=0.
          wy2=0.
          wz2=0.
          do 61 l=1,15
          atest=0.
          h0=twopi*float(l-1)-phi5
          nnn=1
          if(l.eq.1) nnn=10
```

```
      do 62 nb=1,nblds
      do 62 jj=nnn,36
      sp=sn(jj,nb)
      cp=cn(jj,nb)
      f=etap-etav*cp
      h=(h0+phi10*jj)*wm2t
      r0=sqrt(f*f+(etav*sp)**2+h*h)
      fu=ff/(r0*r0*r0)
      atest=atest+fu*(etav-etap*cp)
      wy2=wy2+fu*(tglam*f-h*sp)
      wx2=wx2+fu*(etav*sp*tglam-h*cp)
62    continue
      wz2=wz2+atest
      if(l.gt.1.and.abs(atest).lt.eps) goto 63
61    continue
63    continue
      wxc(nn)=wxc(nn)+wx2
      wyct(nn)=wyct(nn)+wy2
      wzct(nn)=wzct(nn)+wz2
60    continue
c   *
c   *  CONVERGENCE TEST
c   *
      do 70 n=1,nncr2
      if(abs(wzct(n)-wzc(n)).ge.e) goto 71
      if(abs(wyct(n)-wyc(n)).ge.e) goto 71
70    continue
      goto 80
c   *
c   *  CALCULATION OF THE NEXT APPROXIMATION
c   *
71    continue
      wi=0.
      do 55 i=1,nncr2
      wi=wi+wzc(i)
      if(niter.le.3) goto 54
      x0=(wzct(i)*sav1(i)-sav2(i)*wzc(i))/
     &(sav1(i)-wzc(i)-sav2(i)+wzct(i))
      facv(i)=.5*(((x0-sav2(i))/(sav1(i)-sav2(i)))+facv(i))
      if(facv(i).lt..1) facv(i)=.1
      if(facv(i).gt..9) facv(i)=.9
54    continue
      sav1(i)=wzc(i)
      sav2(i)=wzct(i)
      facn=1.-facv(i)
      wyc(i)=wyc(i)*facv(i)+wyct(i)*facn
      wzc(i)=wzc(i)*facv(i)+wzct(i)*facn
55    continue
      wi=abs(wi)/nncr2
      do 74 n=2,nnvr1
      wyv(n)=wyc(n-1)+(wyc(n)-wyc(n-1))*.5
      wzv(n)=wzc(n-1)+(wzc(n)-wzc(n-1))*.5
74    continue
      wyv(1)=wyc(1)
      wyv(nnvr)=wyc(nncr2)
      wzv(1)=wzc(1)*.5
      wzv(nnvr)=wzc(nncr2)*.5
c   *
c   *
      if(lpsem.lt.2) goto 302
```

```
          write(iwr,140)niter
140       format(//," *** SEMRI *** NITER=",i3)
          write(iwr,110) lwx, (wxc(n),n=1,nncr2)
          write(iwr,110) lwy, (wyct(n),n=1,nncr2)
          write(iwr,110) lwz, (wzct(n),n=1,nncr2)
          write(iwr,110) lgamma, (gammc(n),n=2,nncr1)
          write(iwr,110) lfac, (facv(n),n=1,nncr2)
          write(iwr,110) lwy, (wyc(n),n=1,nncr2)
          write(iwr,110) lwz, (wzc(n),n=1,nncr2)
302       continue
c    *
400       continue
c    *
c    *  CONVERGENCE IS NOT REACHED WITHIN LIMS ITERATIONS
c    *  RETURN WITH ITEST1=1
c    *
          itest=1
          write(iwr,102)
102       format(" *** SEMRI *** NO CONVERGENCE ")
80        continue
c    *
          if(ipsem.eq.0) return
c    *
c    *  OUTPUT SECTION
c    *
          lt=0.
          lp=0.
          do 500 n=1,nncr2
          cd=cd0+cdk*alpha(n)**2
          if(abs(alpha(n)).ge.alphas) cd=cd+cd0
          lip(n)=u(n)*gammc(n+1)
          tlip(n)=lip(n)*cos(flamda(n))
          flip(n)=-lip(n)*sin(flamda(n))
          dp(n)=u(n)**2*(pi*sigma/blades)*cd*.5
          fdp(n)=dp(n)*cos(flamda(n))
          tdp(n)=dp(n)*sin(flamda(n))
          tp(n)=tlip(n)+tdp(n)
          fp(n)=flip(n)+fdp(n)
          lt=lt+tp(n)*(etanv(n+1)-etanv(n))
          lp=lp+fp(n)*(etanv(n+1)**2-etanv(n)**2)
500       continue
          lt=lt*blades/pi
          lp=lp*blades/twopi
          if(ifmu.eq.0) go to 5
          cp=lp/fmu**3
          ct=lt/fmu**2
          ctcp=ct/cp
5         continue
          write(iwr,109)
109       format(1h1)
          write(iwr,103) niter
103       format(10x,"SEMI-RIGID WAKE ANALYSIS - CONVERGED AFTER",i4,
         &" ITERATIONS ")
          write(iwr,120)
          write(iwr,115) leta, (etanc(n),n=2,nncr1)
          write(iwr,120)
          write(iwr,110) lwx, (wxc(n),n=1,nncr2)
          write(iwr,110) lwy, (wyct(n),n=1,nncr2)
          write(iwr,110) lwz, (wzct(n),n=1,nncr2)
          write(iwr,110) lu, (u(n),n=1,nncr2)
```

```
          write(iwr,120)
          write(iwr,110)llamda,(flamda(n),n=1,nncr2)
          write(iwr,110)lthet,(thetac(n),n=1,nncr2)
          write(iwr,110)lalpha,(alpha(n),n=1,nncr2)
          write(iwr,110)lgamma,(gammc(n),n=2,nncr1)
          write(iwr,120)
          write(iwr,110)llip,(lip(n),n=1,nncr2)
          write(iwr,110)ltlip,(tlip(n),n=1,nncr2)
          write(iwr,110)lflip,(flip(n),n=1,nncr2)
          write(iwr,120)
          write(iwr,110)ldp,(dp(n),n=1,nncr2)
          write(iwr,110)ltdp,(tdp(n),n=1,nncr2)
          write(iwr,110)lfdp,(fdp(n),n=1,nncr2)
          write(iwr,120)
          write(iwr,110)ltp,(tp(n),n=1,nncr2)
          write(iwr,110)lfp,(fp(n),n=1,nncr2)
          if(ifmu.ne.0) write(iwr,130)lt,lp,ct,cp,ctcp
          if(ifmu.eq.0) write(iwr,130)lt,lp
130       format(//,10x,"LT= ",f10.5,5x,"LP= ",f10.5,/,10x,"CT= ",f10.5,5x,
          &"CP=&",f10.5,/,10x,"CT/CP=&",f10.5)
          return
c    *
115       format(1x,a4,1x,9f10.6,/,6x,9f10.6)
110       format(1x,a4,1x,9f10.6,/,6x,9f10.6)
120       format(1x)
c    *
          end
```

```
      subroutine compa2
c   *
c   *  ********************
c   *  *                  *
c   *  *   PROGRAM FWC    *
c   *  *                  *
c   *  *  SUBROUTINE COMPA2  *
c   *  *                  *
c   *  ********************
c   *
c   *  04/24/78
c   *
c   *  INITIALISATION OF THE POSITIONS OF THE NODES
c   *  AND OF THE INDUCED VELOCITIES,
c   *  FROM THE RESULTS OF SEMRI
c   *
c   *  IF THE NUMBER OF ROLLED UP VORTICES IS 3 THEN NRL22,IRL22,CG2,Q2,
c   *  ARE NOT USED
c   *
      common/lamb/wis
      common/roll6/q1,q2,q3,q4,iw
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,that0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeffi,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nr11,nr12,ir11,ir12,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      dimension wzn(28),wzi(9),dgm1(25),dgm2(25),dgm3(25),dgm4(25),
     &dgm5(25),dgm6(25)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
c   *
      wzc(1)=.8*wzc(1)
      wzc(2)=.8*wzc(2)
      nncr3=nncr2-1
      wzc(nncr2)=.8*wzc(nncr2)
      wzc(nncr3)=.8*wzc(nncr3)
      etam=(etanv(1)+etanv(nnvr))*.5
      do 40 i=2,nncr
      if(etanc(i).gt.etam) goto 41
40    ik=i
41    if(etanc(ik+1)-etam.lt.etam-etanc(ik))ik=ik+1
      wt=wyc(ik+1)
      cpsi=1.+wt/etanc(ik)
c   *
c   *  MEAN INDUCED VELOCITY
c   *
      wzm=0.
      do 1 i=1,nncr2
```

```
          wzm=wzm+wzc(i)*(etanv(i+1)**2-etanv(i)**2)
1         continue
          wzm=wzm/(1.-etanv(1)**2)
c    *
c    *  APPROXIMATION OF THE EXPANSION OF THE WAKE
c    *
          r3=sqrt((fmu+wzm)/(fmu+wzm*2.))
          drdz=wxc(nncr2)/(fmu+wzc(nncr2))
          am=drdz/(r3-1.)
          wwz=wzm+fmu
          if(itrace.ne.0) write(iwr,100)wzm,r3
100       format(1h1//" *** COMPA ***: WZM=",f9.6," R3=",f9.6)
c    *
c    *  NODES AND VELOCITIES FOR THE NEAR WAKE
c    *
          do 20 i=2,nnvr1
          wzn(i)=fmu+wzc(i-1)+(wzc(i)-wzc(i-1))*(etanv(i)-etanc(i))/
           &(etanc(i+1)-etanc(i))
20        continue
          wzn(1)=fmu+wzc(1)*.5
          wzn(nnvr)=fmu+wzc(nncr2)*.5
          do 21 j=1,nnva
          psi=float(j-1)*dpsin*cpsi
          psic=float(j-1)*dpsin
          z=psi*wwz/cpsi
          vr=drdz*exp(-am*z)*wwz
          if(imodel.eq.4) go to 101
          aexp=r3-(r3-1.)*exp(-am*z)
          go to 102
101       aexp=1.0
          ccc=cos(psic)*1.
          sss=sin(psic)*1.
          go to 103
102       ccc=cos(psi)*aexp
          sss=sin(psi)*aexp
103       do 21 i=1,nnvr
          wrnv(i,j)=etanv(i)*vr
          wtnv(i,j)=wt
          wznv(i,j)=wzn(i)-fmu
          xnv(i,j)=ccc*etanv(i)
          ynv(i,j)=sss*etanv(i)
          znv(i,j)=psi*wzn(i)
21        continue
          if(icont.ne.3) goto 90
          write(iwr,500)
500       format(//," COMPA:  WAKE GEOMETRY (Z ONLY)",//)
          do 91 j=1,nnva
          write(iwr,505) j
          write(iwr,510) (znv(i,j), i=1,nnvr)
91        continue
90        continue
505       format(/,i4)
510       format(1x,9f10.6,/,5x,9f10.6)
c    *
c    *  NODES AND VELOCITIES FOR THE INTERMEDIATE WAKE
c    *
          do 30 i=1,nivr
          et=etaiv(i)
          do 31 i2=2,nnvr
          i1=i2-1
```

```
              if(et.lt.etanv(i2)) goto 32
31            continue
32            continue
              a=(etanv(i1+1)-et)/(etanv(i1+1)-etanv(i1))
              ziv(i,1)=a*znv(i1,ntva)+(1.-a)*znv(i1+1,ntva)
              wzi(i)=a*wzn(i1)+(1.-a)*wzn(i1+1)
30            continue
              psiib=dpsin*(ntca-2)
              do 22 j=1,niva
              psi=(psiib+float(j-1)*dpsii)*cpsi
              z=psi*wz/cpsi
              vr=drdz*exp(-am*z)*wz
              aexp=r3-(r3-1.)*exp(-am*z)
              ccc=cos(psi)*aexp
              sss=sin(psi)*aexp
              do 22 i=1,nivr
              wriv(i,j)=etaiv(i)*vr
              wtiv(i,j)=wt
              wziv(i,j)=wzi(i)-fmu
              xiv(i,j)=ccc*etaiv(i)
              yiv(i,j)=sss*etaiv(i)
              ziv(i,j)=ziv(i,1)+(j-1)*dpsii*wzi(i)
22            continue
              if(icont.ne.3) goto 92
              do 93 j=1,niva
              write(iwr,505) j
              write(iwr,510) (ziv(i,j), i=1,nivr)
93            continue
92            continue
              nncr5=nncr2+3
c             COMPUTE THE STRENGTH OF TIP,ROOT & MIDDLLE VORTICES
              do 1000 j=3,nncr2
              k=nncr5-j
              if(abs(gammc(k)).lt.abs(gammc(k-1))) go to 1000
              go to 2000
1000          continue
2000          q4=-gammc(k)
              q3=gammc(k)-gammc(k-3)
              q1=gammc(k-3)-gammc(2)
c             LOCATION OF SECOND MIDDLE VORTEX
              k3=k-3
              k4=k-4
c             p=0.
c             q=0.
c             k1=k-1
c             do 3000 l=4,k4
c             dgm1(l+1)=(gammc(l+1)-gammc(l))*etanv(l)**2
c             p=p+dgm1(l+1)
c             q=q+(gammc(l+1)-gammc(l))*etanv(l)
c 3000        continue
c             cg2=p/q
c
c             LOCATION OF TIP VORTEX
              p1=0.
              p2=0.
              do 4000 i=k,nncr1
              dgm3(i+1)=(gammc(i)-gammc(i+1))*etanv(i)**2
              p1=dgm3(i+1)+p1
              p2=p2+(gammc(i)-gammc(i+1))*etanv(i)
4000          continue
```

```
          cg4=p1/p2
          c5=0.0
c   *
c   *     LOCATION OF FIRST MIDDLE VORTEX
          k3=k-3
          b=0.0
          do 7000 j=k3,k1
          a=(gammc(j+1)-gammc(j))*etanv(j)**2
          b=b+a
          c5=c5+(gammc(j+1)-gammc(j))*etanv(j)
 7000     continue
          cg3=b/c5
*
c*        LOCATION OF ROOT VORTEX
c*
          pp1=0.
          pp2=0.
          k1=k-1
          k4=k-4
          gammc(1)=0.0
          do 8000 i=2,k4
          pp1=pp1+(gammc(i+1)-gammc(i))*etanv(i)**2
          pp2=pp2+(gammc(i+1)-gammc(i))*etanv(i)
 8000     continue
          cg1=pp1/pp2
          nr121=nr12-1
c         nr122=nr12-2
          ir121=ir12-1
c         ir122=ir12-2
          do 5002 i=1,nnvr
          if(etanv(i).gt.cg1) go to 5003
5002      ic=i
5003      tau5=cg1-etanv(ic)
          tau6=etanv(ic+1)-etanv(ic)
          wzn(nr11)=wzn(ic)+(wzn(ic+1)-wzn(ic))*(tau5/tau6)
c         do 5004 i=1,nnvr
c         if(etanv(i).gt.cg2) go to 5006
c 5004    id=i
c 5006    tau7=cg2-etanv(id)
c         tau8=etanv(id+1)-etanv(id)
c         wzn(nr122)=wzn(id)+(wzn(id+1)-wzn(id))*(tau7/tau8)
          do 7001 i=1,nnvr
          if(etanv(i).gt.cg3) go to 7002
 7001      ie=i
 7002      tau9=cg3-etanv(ie)
          tau10=etanv(ie+1)-etanv(ie)
          wzn(nr121)=wzn(ie)+(wzn(ie+1)-wzn(ie))*(tau9/tau10)
          do 8001 i=1,nnvr
          if(etanv(i).gt.cg4) go to 8002
8001      iq=i
8002      tau11=cg4-etanv(iq)
          tau12=etanv(iq+1)-etanv(iq)
          wzn(nr12)=wzn(iq)+(wzn(iq+1)-wzn(iq))*(tau11/tau12)
          do 5007 j=1,nnva
          psi=float(j-1)*dpsin*cpsi
          z=psi*wwz/cpsi
          vr=drdz*exp(-am*z)*wwz
          aexp=r3-(r3-1.)*exp(-am*z)
          ccc=cos(psi)*aexp
          sss=sin(psi)*aexp
```

```
      do 5007 i=nrl1,nrl2
      if(i.eq.nrl1) roll=cg1
c     if(j.eq.nrl22) roll=cg2
      if(i.eq.nrl21) roll=cg3
      if(i.eq.nrl2) roll=cg4
      wrnv(i,j)=roll*vr
      wtnv(i,j)=wt
      wznv(i,j)=wzn(i)-fmu
      xnv(i,j)=ccc*roll
      ynv(i,j)=sss*roll
      znv(i,j)=psi*wzn(i)
5007  continue
      if(icont.ne.3) goto 94
      do 95 j=1,nnva
      write(iwr,505) j
      write(iwr,510) (znv(i,j), i=nrl1,nrl2)
95    continue
94    continue
      wzi(irl1)=wzn(nrl1)
      wzi(irl2)=wzn(nrl2)
      wzi(irl21)=wzn(nrl21)
c     wzi(irl22)=wzn(nrl22)
      ziv(irl1,1)=znv(nrl1,ntva)
      ziv(irl2,1)=znv(nrl2,ntva)
      ziv(irl21,1)=znv(nrl21,ntva)
c     ziv(irl22,1)=znv(nrl22,ntva)
      do 10 j=1,niva
      psi=(psiib+float(j-1)*dpsii)*cpsi
      z=psi*wwz/cpsi
      vr=drdz*exp(-am*z)*wwz
      aexp=r3-(r3-1.)*exp(-am*z)
      ccc=cos(psi)*aexp
      sss=sin(psi)*aexp
      do 5008 i=irl1,irl2
      if(i.eq.irl1) rolll=cg1
c     if(i.eq.irl22) rolll=cg2
      if(i.eq.irl21) rolll=cg3
      if(i.eq.irl2) rolll=cg4
      wriv(i,j)=rolll*vr
      wtiv(i,j)=wt
      wziv(i,j)=wzi(i)-fmu
      xiv(i,j)=ccc*rolll
      yiv(i,j)=sss*rolll
      ziv(i,j)=ziv(i,1)+(j-1)*dpsii*wzi(i)
5008  continue
10    continue
      if(icont.ne.3) goto 96
      do 97 j=1,niva
      write(iwr,505) j
      write(iwr,510) (ziv(i,j), i=irl1,irl2)
97    continue
96    continue
c*
c*    self induced velocity
      c1=0.
      c2=0.
      r1=.95
      r2=1.05
      do 1009 kw=1,2
      do 1009 i=1,180
```

```
      psr=i*0.03142+0.3142
      if(kw.eq.2) go to 1010
      b=1-r1*cos(psr)
      d=(1+r1**2-2*r1*cos(psr))**1.5
      c1=c1+(b/d)*0.03142
      go to 1009
1010  b=1-r2*cos(psr)
      d=(1+r2**2-2*r2*cos(psr))**1.5
      c2=c2+(b/d)*0.03142
1009  continue
      wi=(c1+c2)/2.
c*    Lamb induced velocity (vortex ring) with d=.01,D=2
      wil=5.52
      wis=wil-wi
      return
      end
```

```
      subroutine inivnd2
c   *
c   *  ***********************
c   *  *                     *
c   *  *     PROGRAM FWC     *
c   *  *                     *
c   *  *  SUBROUTINE INIVND  *
c   *  *                     *
c   *  ***********************
c   *
c   *  02/20/78
c   *
c   *  PARAMETERS FOR THE EVALUATION OF THE STRENGH OF THE ELEMENTS
c   *  FROM THE DISTRIBUTION OF CIRCULATION ALONG THE BLADES
c   *
c   *  FOR EACH ELEMENT (I)
c   *  HM1(I,J) IS THE STRENGH OF THE ELEMENT ON THE INNER SIDE
c   *  MULTIPLIED BY THE SIDE LENGTH OF THE ELEMENT, FOR A UNIT
c   *  CIRCULATION AT THE INNER CENTER (J) OF THE BLADE
c   *  HM2(I,J) IS THE STRENGH OF THE ELEMENT ON THE OUTER SIDE
c   *  MULTIPLIED BY THE SIDE LENGTH OF THE ELEMENT, FOR A UNIT
c   *  CIRCULATION AT THE INNER CENTER (J) OF THE BLADE
c   *
      common/cvind/hm1n(24,24),hm2n(24,24),hammt1(24),hammt2(24),hammt3(24)
      common/parm/iwr,ird,itrace,ipscm,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel.
      dimension gnc(30),gic(7),hntv(24),hitv(6),gntv(25),gitv(24)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      data lh1i,lh2i/"HM1I","HM2I"/
      data lh1n,lh2n/"HM1N","HM2N"/
      data lht1,lht2,lht3,lht4/"HMT1","HMT2","HMT3","HMT4"/
c   *
c   *  CLEAR ARRAY OF UNIT CIRCULATION ALONG THE BLADES
c   *
      do 10 i=1,nncr
      gnc(i)=0.
10    continue
c   *
c   *  LOOP ON THE UNIT CIRCULATIONS ALONG THE BLADES
c   *
      nncr3=nncr2-1
      do 12 j=1,nncr3
      gnc(j+1)=1.
c   *
c   *  LUMPED STRENGH OF THE TRAILING VORTICES
c   *
      do 14 i=1,nnvr1
      gntv(i)=gnc(i+1)-gnc(i)
14    continue
c   *
c   *  DISTRIBUTED STRENG OF THE TRAILING VORTICES
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
c    *
           nnvr2=nnvr1-1
           do 15 i=2,nnvr2
           hntv(i)=gntv(i)/(etanc(i+1)-etanc(i))
15         continue
           ht1=gntv(1)*coeff
           ht2=gntv(nnvr1)*coeff
           hntv(1)=2.*(gntv(1)-ht1)/(etanc(2)-etanc(1))
           hntv(nnvr1)=2.*(gntv(nnvr1)-ht2)/(etanc(nncr1)-etanc(nncr2))
c    *
c    *  COEFFICIENTS FOR THE NEAR WAKE
c    *
           hammt1(j)=ht1
           hammt2(j)=ht2
           do 18 i=1,nnvr2
           hm1n(i,j)=hntv(i)*(etanv(i+1)-etanv(i))
           hm2n(i,j)=hntv(i+1)*(etanv(i+1)-etanv(i))
18         continue
c*
c    *
           gnc(j+1)=0.
12         continue
c    *
c    *
c    *
c    ********************
c    *
           if(itrace.eq.0)return
           write(iwr,150)
150        format(" *** INIVIND ***")
           write(iwr,151)
151        format(/,"  INT.COEFF.  (NEAR WAKE)")
           do 68 ki=1,nnvr1
           write(iwr,110)lh1n,(hm1n(ki,kj),kj=1,nncr2)
           write(iwr,110)lh2n,(hm2n(ki,kj),kj=1,nncr2)
68         continue
           write(iwr,110)lht1,(hammt1(kj),kj=1,nncr2)
           write(iwr,110)lht2,(hammt2(kj),kj=1,nncr2)
           write(iwr,152)
152        format(/,"  INT COEFF.  (INTERMEDIATE WAKE)")
           return
c    *
110        format(1x,a4,1x,9f13.8,/,6x,9f13.8)
120        format(1x,a4,1x,9i13,/,6x,9i13)
c    *
           end
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      subroutine cofn2
c   *
c   * ********************
c   * *                  *
c   * *  PROGRAM FWC     *
c   * *                  *
c   * *  SUBROUTINE COFN *
c   * *                  *
c   * ********************
c   *
c   *      02/03/78
c   *
c   * EVALUATION OF THE COORDINATES OF THE CENTERS
c   * GIVEN THE COORDINATES OF THE NODES
c   *
      common/roll3/etar(4)
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &ltwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeffi,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nr11,nr12,ir11,ir12,imodel
      common /resul/gamma(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),
     &zic(9,51),xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),
     &yiv(8,50),ziv(8,50),ia,ib
      dimension rnvl(19),rnvr(19),rivl(51),rivr(51),dpnl(18),dpnr(18),
     &dpil(50),dpir(50)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
c   *
c   * INNER POINTS
c   *
      do 1 i=2,nncr2
      do 1 j=2,nnca1
      xnc(i,j)=.25*(xnv(i-1,j-1)+xnv(i-1,j)+xnv(i,j-1)+xnv(i,j))
      ync(i,j)=.25*(ynv(i-1,j-1)+ynv(i-1,j)+ynv(i,j-1)+ynv(i,j))
      znc(i,j)=.25*(znv(i-1,j-1)+znv(i-1,j)+znv(i,j-1)+znv(i,j))
1     continue
c   *
c   * CENTERS OUTSIDE THE WAKE(EXEPT CORNERS)
c   *
      do 3 i=2,nncr2
      xnc(i,1)=xnv(i,1)+xnv(i-1,1)-xnc(i,2)
      ync(i,1)=ynv(i,1)+ynv(i-1,1)-ync(i,2)
      znc(i,1)=znv(i,1)+znv(i-1,1)-znc(i,2)
      xnc(i,nnca)=xnv(i,nnva)+xnv(i-1,nnva)-xnc(i,nnca1)
      ync(i,nnca)=ynv(i,nnva)+ynv(i-1,nnva)-ync(i,nnca1)
      znc(i,nnca)=znv(i,nnva)+znv(i-1,nnva)-znc(i,nnca1)
3     continue

      do 7 j=1,nnca1
      rnvl(j)=sqrt(xnv(1,j)**2+ynv(1,j)**2)
      rnvr(j)=sqrt(xnv(nnvr1,j)**2+ynv(nnvr1,j)**2)
      if(j.eq.nnca1) go to 7
      dpnl(j)=atan2((ynv(1,j)+ynv(1,j+1)),(xnv(1,j)+xnv(1,j+1)))
      dpnr(j)=atan2((ynv(nnvr1,j)+ynv(nnvr1,j+1)),(xnv(nnvr1,j)+xnv(nnvr1,j+
```

```
     &1)))
7    continue
     do 6 j=2,nncal
     xnc(1,j)=(rnvl(j)+rnvl(j-1))*cos(dpnl(j-1))-xnc(2,j)
     ync(1,j)=(rnvl(j)+rnvl(j-1))*sin(dpnl(j-1))-ync(2,j)
     znc(1,j)=znv(1,j)+znv(1,j-1)-znc(2,j)
     xnc(nncr1,j)=(rnvr(j)+rnvr(j-1))*cos(dpnr(j-1))-xnc(nncr2,j)
     ync(nncr1,j)=(rnvr(j)+rnvr(j-1))*sin(dpnr(j-1))-ync(nncr2,j)
     znc(nncr1,j)=znv(nnvr1,j)+znv(nnvr1,j-1)-znc(nncr2,j)
6    continue
c  *
c  * CORNERS
c  *
     xnc(1,1)=4.*xnv(1,1)-xnc(1,2)-xnc(2,1)-xnc(2,2)
     ync(1,1)=4.*ynv(1,1)-ync(1,2)-ync(2,1)-ync(2,2)
     znc(1,1)=4.*znv(1,1)-znc(1,2)-znc(2,1)-znc(2,2)
     xnc(nncr1,1)=4*xnv(nnvr1,1)-xnc(nncr1,2)-xnc(nncr2,1)-xnc(nncr2,2)
     ync(nncr1,1)=4*ynv(nnvr1,1)-ync(nncr1,2)-ync(nncr2,1)-ync(nncr2,2)
     znc(nncr1,1)=4*znv(nnvr1,1)-znc(nncr1,2)-znc(nncr2,1)-znc(nncr2,2)
     xnc(1,nnca)=4.*xnv(1,nnva)-xnc(2,nnca)-xnc(1,nncal)-xnc(2,nncal)
     ync(1,nnca)=4.*ynv(1,nnva)-ync(2,nnca)-ync(1,nncal)-ync(2,nncal)
     znc(1,nnca)=4.*znv(1,nnva)-znc(2,nnca)-znc(1,nncal)-znc(2,nncal)
     xnc(nncr1,nnca)=4.*xnv(nnvr1,nnva)-xnc(nncr2,nnca)-xnc(nncr1,nncal)
     &-xnc(nncr2,nncal)
     ync(nncr1,nnca)=4.*ynv(nnvr1,nnva)-ync(nncr2,nnca)-ync(nncr1,nncal)
     &-ync(nncr2,nncal)
     znc(nncr1,nnca)=4.*znv(nnvr1,nnva)-znc(nncr2,nnca)-znc(nncr1,nncal)
     &-znc(nncr2,nncal)
c  *
c  *
c *
     return
     end
```

```
        subroutine loop22
c    *
c    *  ***********************
c    *  *                     *
c    *  *    PROGRAM EHC      *
c    *  *                     *
c    *  *  SUBROUTINE LOOP22  *
c    *  *                     *
c    *  ***********************
c    *
c    *  12/1/80
c    *
c    *  LOOP ON THE ELEMENTS, TO COMPUTE THE VELOCITIES
c    *  AT THE CENTERS, BY VINDB,H,I
c    *
c    *  IF THE NUMBER OF ROLLED UP VORTICES IS 3 THEN IRL22,NRL22,CG2,Q2
c    *  ARE NOT USED
c    *  BLADES            BL
c    *  NEAR WAKE         /  A /  B /    /    /
c    *  INT WAKE          /    / B' /  C /    /
c    *  FAR WAKE          /    /    /    /  F /
c    *
        common/roll6/q1,q2,q3,q4,iw
        common/all/cg1,cg2,cg3,cg4,ir
        common/roll10/wxctr2(25),wyctr2(25),wzctr2(25)

        common/parm/iwr,ird,itrace,ipsem,nmas,itrcw,itrcl1,itrcl2,itrcg,
         &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamot,
         &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
         &ipr,iplott
        common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
         &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
         &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
         &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
         &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
         &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel
        common /resul/gamma(26),twx(24,24),twy(24,24),twz(24,24),
         &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
         &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
         &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
         &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
         &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
         &ia,ib
        common/cvind/hm1n(24,24),hm2n(24,24),hammt1(24),hammt2(24),hammt3(24)
        common/roll1/hammt5(26),hammt6(26)
        common /vel/x,y,z,ux,uy,uz,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,
         &gm,gm1,gm2,dgm,it,rho,zpf,t1,eps1,eps2,stre,jgm
        common /vort/gm1n(24),gm2n(24),gm1i(5),gm2i(5),
         &gammt1,gammt2,gammt3,gammt4
        common /savtw/twyt(24,24),twzt(24,24)
        common/tipco/wxnt1,wynt1,wznt1,wxnt2,wynt2,wznt2,wxntp,wyntp,wzntp
        common/semi/delz
        equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
        equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
        dimension dgm1(25),dgm2(25),dgm3(25),dgm4(25),dgm5(15),dgm6(15)
c    *
c    *  S2=SIN(15),C2=COS(15),TT2=2.*TG(15/2)
        data s2,c2,tt2/.258819045,.965925826,.263304995/
        data twopi/6.283185306/
        data lg1n,lg2n,lg1i,lg2i/"GM1N","GM2N","GM1I","GM2I"/
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      data lgam/"GAMC"/
c   *
c   ****************************************
c   *
c   *  STRENGH OF THE ELEMENTS
c   *
c          TIP  LOSS
c
         gammc(nncr1)=0.0
      nnvr2=nnvr1-1
      nncr3=nncr2-1
      do 600 i=1,nnvr1
      gm1n(i)=0.
      gm2n(i)=0.
      do 600 j=1,nncr2
      gm1n(i)=gm1n(i)+hm1n(i,j)*gammc(j+1)
      gm2n(i)=gm2n(i)+hm2n(i,j)*gammc(j+1)
600   continue
c   *
c   *
      gammt1=0.
      gammt2=0.
      do 604  j=1,nncr2
      gammt1=gammt1+hammt1(j)*gammc(j+1)
      gammt2=gammt2+hammt2(j)*gammc(j+1)
604   continue
      jgm=0
      if(nmes.ne.1) go to 723
      gtmin=0.
      gtmax=0.
      if(gtmin.gt.gammc(j+1))  gtmin=gammc(j+1)
      if(gtmax.lt.gammc(j+1))  gtmax=gammc(j+1)
      if(gtmin.gt.gammc(j+1))  jmin=j
      if(gtmax.lt.gammc(j+1))  jmax=j
725   continue
      gtt=gtmax
      if(-gtmin.gt.gtmax)  gtt=gtmin
      jtm=jmax
      if(-gtmin.gt.gtmax)  jtm=jmin
723   continue
c   *
      if(itrcg.eq.0)  goto 610
      write(iwr,153)  niter
153   format(///,"*** LOOP2 *** (ITERATION:",I3,")")
      write(iwr,110) lgam, (gammc(ki),ki=2,nncr1)
      if(itrcl2.eq.0)  goto 610
      write(iwr,110) lg1n, (gm1n(ki),ki=1,nnvr1)
      write(iwr,110) lg2n, (gm2n(ki),ki=1,nnvr1)
      write(iwr,110) lg1i, (gm1i(ki),ki=1,nivr1)
      write(iwr,110) lg2i, (gm2i(ki),ki=1,nivr1)
      write(iwr,615) gammt1,gammt2,gammt3,gammt4
615   format(" GAMMT 1 TO 4 ",4e13.4)
110   format(1x,a4,1x,9f10.6,/,6x,9f10.6)
610   continue
c   *
c   ******************************************
c   *
c   *  SAVE INFLUENCE COEFFICIENTS OF PRECEDENT ITERATION
c   *
      do 60 i=1,nncr2
```

```
       do 60 j=1,nncr2
       twyt(i,j)=twy(i,j)
       twzt(i,j)=twz(i,j)
60     continue
c  *
c  *********************************************
c  *
       eps1=fps1
       eps2=fps2
       itt2=itrcl2
       if(itt2.eq.4) itrcl2=2
       if(itref.eq.1)itrcl2=0
999    continue
       call clv2
       ideb=5
       if(itrans.eq.0.or.itref.eq.1)ideb=1
       if(itref.eq.1) goto 250
       if(itrent.eq.1) call inicn2
       if(itrcl2.ne.0) write(iwr,200)
200    format("1*** LOOP2 ***")
c  *
       if(imodel.eq.3) go to 111
       if(imodel.eq.4) go to 222
       go to 333
111    do 444 i=1,nnvr1
       do 444 j=1,nnva
444    znv(i,j)=0.0
       go to 333
222    do 2111 i=1,nnvr1
       do 2111 j=1,nnva
2111   znv(i,j)=znv(nr12,j)
       do 2222 i=nr11,nr12
       do 2222 j=1,nnva
2222   znv(i,j)=znv(nr12,j)
c  *
333    continue
c  *
c  *
c  *******************
c  *
c  *  VELOCITIES INDUCED BY THE CIRCULATION ON THE BLADES
c  *
c      TIP LOSS FACTOR
c
       gamma(nncr1)=0.0
       it=2
       iw=1
       if(itrcl2.gt.1) write(iwr,301)
301    format(" VELOCITIES INDUCED BY THE CIRCULATION ON THE BLADES")
       x1=etanv(1)
       y1=0.
       z1=0.
       y3=0.
       z3=0.
       do 1 i=1,nncr2
       x3=etanv(i+1)
       gm=-gamma(i+1)
       call coord
       call vindn12(1,nnva)
       call vindi2(2,niva)
```

```
          if(imodel.eq.2) call vindn2(1,nnca)
          x1=x3
 1        continue
          if(itrcl2.eq.2) call prtv
          if(itrcl2.eq.3) call prtvd2
c    *
          wxnt1=wxnc(nnvr1,2)
          wynt1=wync(nnvr1,2)
          wznt1=wznc(nnvr1,2)
          wxnt2=wxnc(nncr1,2)
          wynt2=wync(nncr1,2)
          wznt2=wznc(nncr1,2)
          wxntp=(wxnc(nnvr1,1)+wxnc(nncr1,1)+wxnc(nnvr1,2)+wxnc(nncr1,2))/4.
          wyntp=(wync(nnvr1,1)+wync(nncr1,1)+wync(nnvr1,2)+wync(nncr1,2))/4.
          wzntp=(wznc(nnvr1,1)+wznc(nncr1,1)+wznc(nnvr1,2)+wznc(nncr1,2))/4.
c    ********************
c    *
c    *  VELOCITIES INDUCED BY THE NEAR WAKE
          phib=twopi/blades
          kapa=ifix(phib/dpsi1)
          nvsp=niva-kapa
c    *
c    *  SURFACE ELEMENTS
          it=1
          if(itrcl2.gt.1) write(iwr,302)
302       format("1VELOCITIES INDUCED BY THE NEAR WAKE (SURFACE ELEMENTS)")
          do 3 i=1,nnvr2
          gm1=gm1n(i)
          gm2=gm2n(i)
          do 3 j=1,nnva1
          x1=xnv(i,j)
          y1=ynv(i,j)
          z1=znv(i,j)
          x2=xnv(i+1,j)
          y2=ynv(i+1,j)
          z2=znv(i+1,j)
          x3=xnv(i,j+1)
          y3=ynv(i,j+1)
          z3=znv(i,j+1)
          x4=xnv(i+1,j+1)
          y4=ynv(i+1,j+1)
          z4=znv(i+1,j+1)
          call coord
          call vindb2(1,1)
          if(imodel.eq.4) it=8
          if(imodel.eq.4) call vindn12(2,nnva)
          if(imodel.eq.4) it=1
          if(imodel.eq.2) call vindn2(1,nnca)
3         continue
          if(itrcl2.eq.2) call prtv
          if(itrcl2.eq.3) call prtvd2
          if(itrcnt.eq.1) call execn2
c    *
c    *  SEGMENTS ELEMENTS
c    *
          it=2
          if(itrcl2.gt.1) write(iwr,303)
303       format("1VELOCITIES INDUCED BY THE NEAR WAKE (SEG. ELEMENTS)")
          do 500 i=1,nnvr1,nnvr2
          if((nmes.eq.2).and.(i.eq.1)) go to 500
```

```
          gm=gammt1
          if(i.eq.nnvr1) gm=gammt2
          do 510 j=1,nnval
          x1=xnv(i,j)
          y1=ynv(i,j)
          z1=znv(i,j)
          x3=xnv(i,j+1)
          y3=ynv(i,j+1)
          z3=znv(i,j+1)
          call coord
          call vindb2(2,1)
          if(imodel.eq.4) it=9
          if(imodel.eq.4) call vindn12(2,nnva)
          if(imodel.eq.4) it=2
          if(imodel.eq.2) call vindn2(1,nnca)
510       continue
500       continue
c   *
c   **********************
c   *
c   *
c   *  COMPUTE THE STRENGTH OF TIP, MIDDLE,AND ROOT VORTICES
c   *
          it=2
c
          nncr5=nncr2+3
          do 1000 j=3,nncr2
          k=nncr5-j
          if(abs(gammc(k)).lt.abs(gammc(k-1))) go to 1000
          go to 2000
1000      continue
2000      q4=-gammc(k)
          q3=gammc(k)-gammc(k-3)
          q1=gammc(k-3)-gammc(2)
c
c         LOCATION OF SECOND MIDDLE ROLLED UP VORTEX
          k1=k-1
          k3=k-3
          k4=k-4
c         p=0.
c         pp=0.
c         do 4000 l=4,k4
c         dgm1(l+1)=(gammc(l+1)-gammc(l))*etanv(l)
c         p=p+dgm1(l+1)
c         pp=pp+(gammc(l+1)-gammc(l))
c 5000    continue
c         cg2=p/pp
c
c         LOCATION OF THE TIP ROLLED UP VORTEX
          p1=0.
          p2=0.
          do 4000 i=k,nncr2
          dgm3(i+1)=(gammc(i)-gammc(i+1))*etanv(i)
          p1=p1+dgm3(i+1)
          p2=p2+(gammc(i)-gammc(i+1))
4000      continue
          cg4=p1/p2
c*
c*        LOCATION OF THE FIRST MIDDLE ROLLED UP VORTEX
          b=0.0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      c5=0.0
      do 7000 j=k3,k1
      b=b+(gammc(j+1)-gammc(j))*etanv(j)
      c5=c5+(gammc(j+1)-gammc(j))
 7000 continue
      cg3=b/c5
c*
c*    LOCATION OF THE ROOT ROLLED UP VORTEX
      pp1=0.
      pp2=0.
      gammc(1)=0.0
      do 8000 i=2,k4
      pp1=pp1+(gammc(i+1)-gammc(i))*etanv(i)
      pp2=pp2+(gammc(i+1)-gammc(i))
 8000 continue
      cg1=pp1/pp2
c*
      write(iwr,4001) cg1,cg2,cg3,cg4,q1,q2,q3,q4
4001  format(" location of rollup vortices = ",8f10.5)
      if(itrcl2.gt.1) write(iwr,307)
307   format("1VELOCITIES INDUCED BY THE INT. WAKE (SEG. ELEMENTS)")
      ir121=ir12-1
      nr121=nr12-1
c     nr122=nr12-2
c     ir122=ir12-2
c     VELOCITIES INDUCED BY ANTICIPATED NEAR WAKE ROLLED UP VORTICES
c
      do 7 i=nr11,nr12
      if(i.eq.nr11) gm=q1
c     if(i.eq.nr122) gm=q2
      if(i.eq.nr121) gm=q3
      if(i.eq.nr12) gm=q4
      do 6 j=1,nnval
      x1=xnv(i,j)
      y1=ynv(i,j)
      z1=znv(i,j)
      x3=xnv(i,j+1)
      y3=ynv(i,j+1)
      z3=znv(i,j+1)
      call coord
      call vindn12(2,nnva)
      call vindi2(2,niva)
6     continue
7     continue
c
c     VELOCITIES INDUCED BY INTERMEDIATE WAKE ROLLED UP VORTICES
      do 18 i=ir11,ir12
      if((nmes.eq.2).and.(i.eq.1)) go to 18
      if(i.eq.ir11) gm=q1
c     if(i.eq.ir122) gm=q2
      if(i.eq.ir121) gm=q3
      if(i.eq.ir12) gm=q4
      do 17 j=1,nival
      dps2=eps2
      x1=xiv(i,j)
      y1=yiv(i,j)
      z1=ziv(i,j)
      x3=xiv(i,j+1)
      y3=yiv(i,j+1)
c     CORE BURSTING AFTER FIRST BLADE INTERSECTION
```

```
          phil=dpsin*(nnva-1)+dpsil*(j-1)
          if(phil.gt.phib) eps2=.1
          call coord
          call vindnl2(2,nnva)
          call vindi2(2,niva)
          if(imodel.eq.2) call vindn2(1,nnca)
          eps2=dps2
17        continue
18        continue
          if(itrcl2.eq.2) call prtv
          if(itrcl2.eq.3) call prtvd2
          if(itrent.eq.1) call execn2
c    *
c    ***************************
c    *
c    *  VELOCITIES INDUCED BY THE FAR WAKE
c    *
c    *
250       continue
c    *
c    *
c*        FOR 2 SPIRALS : 720 DEGREES
c*
          do 50 i=irl1,irl2
          if(i.eq.irl21) gm=q3*1.
          if(i.eq.irl1) gm=q1*1.
          if(i.eq.irl2) gm=q4*1.
c         if(i.eq.irl22) gm=q2*1.
          if(itrcl2.gt.1) write(iwr,308)i,gm
308       format("1V.L. INDUCED BY THE FAR WAKE(SEG),I=",i3," GM=",e13.4)
          delz=ziv(i,niva)-ziv(i,nvsp)
          zpf=ziv(i,niva)*2-ziv(i,nvsp)
          rho=.5*sqrt((xiv(i,niva)+xiv(i,niva1))**2+
           &(yiv(i,niva)+yiv(i,niva1))**2)
          if(itrcf.ne.1) goto 201
c    *&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
c    *  VERIFICATION OF THE MODEL OF THE FAR WAKE
          nk=480
          do 220 ktt=1,2
          it=4-ktt
          write(iwr,205)it,i
205       format("1** VERIFICATION OF THE FAR WAKE *** IT=",i3," II=",i3)
          if(it.eq.2) goto 210
          call vindnl(1,nnca)
          call vindi(2,niva)
          call prtvd
220       continue
c    *&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
201       continue
c    *
          it=3
          call vindnl2(1,nnva)
          call vindi2(2,niva)
          if(imodel.eq.2) call vindn2(1,nnca)
c    *
c    *
210       continue
50        continue
c    *
          if(itrcl2.eq.2) call prtv
```

```
      if(itrcl2.eq.3) call prtvd2
75    continue
      if(itrcl2.eq.1) write(iwr,310)
310   format(" VELOCITIES INDUCED AT THE CENTERS")
c     if(itrcl2.eq.1) call prtv
      if(itrent.eq.1.and.itrcf.eq.0.and.itrcl2.ne.3) call excen2
      if(itrcf.eq.1) goto 888
      if(itrcl2.eq.3) goto 889
      if(itrcl2.eq.2) itrcl2=1
      if(itt2.eq.4) itrcl2=4
      return
888   continue
      itrcf=0
      itrcl2=itt2
      goto 999
889   continue
      itrcl2=1
      goto 999
      end
```

```
      subroutine prtv
c   *
c   *  *********************
c   *  *                   *
c   *  *   PROGRAM FWC     *
c   *  *                   *
c   *  *   SUBROUTINE PRTV *
c   *  *                   *
c   *  *********************
c   *
c   *   02/03/78
c   *
c   *  PRINT OUT FOR VERIFICATION
c   *  ENTRIES:
c   *  PRTV: VELOCITIES AT THE CENTERS AND INFLUENCE COEFFICIENTS
c   *  PRTCEN: POSITIONS OF THE CENTERS
c   *  PRTNOD: POSITIONS OF THE NODES
c   *  PRTVD  VELOCITIES AT THE CENTERS.
c   *  THE CLEAR THE CORRESPONDING ARRAYS
c   *  CLV  CLEAR THE ARRAYS FOR THE INDUCED VELOCITIES
c   *  AND THE INFLUENCE COEFFICIENTS
c   *  PRTVN  VELOCITIES AT THE CENTERS OF THE NEAR WAKE
c   *  PRTVI  VELOCITIES AT THE CENTERS OF THE INTERMEDIATE WAKE
c   *
      common/rol12/wzivr(8,50),wxivr(8,50),wyivr(8,50),wznvr(28,18),wxnvr(28,18),
     &wynvr(28,18)
      common/rol15/wxctr(25),wyctr(25),wzctr(25)
      common/parm/iwr,ird,itrace,ipsem,nmas,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplocw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      dimension wr(26)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      data lwxnc,lwync,lwznc,lwr/"WXNC","WYNC","WZNC","WR  "/
      data lwxic,lwyic,lwzic    /"WXIC","WYIC","WZIC"/
      data ltwx,ltwy,ltwz/"TWX ","TWY ","TWZ "/
      data lxnc,lync,lznc/"XNC ","YNC ","ZNC "/
      data lxic,lyic,lzic/"XIC ","YIC ","ZIC "/
      data lxnv,lynv,lznv/"XNV ","YNV ","ZNV "/
      data lxiv,lyiv,lziv/"XIV ","YIV ","ZIV "/
c   *
      w(x,y,wx,wy)=(wx*x+wy*y)/sqrt(x**2+y**2)
c   *
101   format(1x,a4,1x,6f10.6,/,6x,6f10.6,/,6x,6f10.6)
102   format(1x)
c   *
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        idel=0
        write(iwr,100)
100     format(" *** PRTV ***")
c   *
60      continue
c   *
        do 1 j=1,nnca
        do 4 i=1,nncr
        wr(i)=w(xnc(i,j),ync(i,j),wxnc(i,j),wync(i,j))
4       continue
        write(iwr,102)
        write(iwr,101)lwxnc,(wxnc(i,j),i=1,nncr)
        write(iwr,101)lwync,(wync(i,j),i=1,nncr)
        write(iwr,101)lwznc,(wznc(i,j),i=1,nncr)
        write(iwr,101)lwr,(wr(i),i=1,nncr)
1       continue
        if(idel.eq.2) return
c   *
61      continue
        do 2 j=1,nica
        do 5 i=1,nicr
        wr(i)=w(xic(i,j),yic(i,j),wxic(i,j),wyic(i,j))
5       continue
        write(iwr,102)
        write(iwr,101)lwxic,(wxic(i,j),i=1,nicr)
        write(iwr,101)lwyic,(wyic(i,j),i=1,nicr)
        write(iwr,101)lwzic,(wzic(i,j),i=1,nicr)
        write(iwr,101)lwr,(wr(i),i=1,nicr)
2       continue
        if(idel.eq.3) return
c   *
        if(idel.eq.1) goto 70
c   *
c   *  PRINT OUT OF THE INFLUENCE COEFFICIENTS
c   *
        do 3 j=1,nncr2
        write(iwr,102)
        write(iwr,101)ltwx,(twx(i,j),i=1,nncr2)
        write(iwr,101)ltwy,(twy(i,j),i=1,nncr2)
        write(iwr,101)ltwz,(twz(i,j),i=1,nncr2)
3       continue
        return
c   *
c   ****************************************************
c   *
        entry prtcen2
c   *
c   *  PRINT THE POSITIONS OF THE CENTERS
c   *
        write(iwr,103)
103     format(" *** PRTCEN ***")
        do 30 j=1,nnca
        write(iwr,102)
        write(iwr,101)lxnc,(xnc(i,j),i=1,nncr)
        write(iwr,101)lync,(ync(i,j),i=1,nncr)
        write(iwr,101)lznc,(znc(i,j),i=1,nncr)
30      continue
        do 32 j=1,nica
        write(iwr,102)
        write(iwr,101)lxic,(xic(i,j),i=1,nicr)
```

```
      write(iwr,101) lyic, (yic(i,j),i=1,nicr)
      write(iwr,101) lzic, (zic(i,j),i=1,nicr)
32    continue
      return
c  *
c  ***********************************************
c  *
      entry prtnod2
c  *
c  *  PRINT THE POSITIONS OF THE NODES
c
      write(iwr,104)
104   format(" *** PRTNOD ***")
      do 50 j=1,nnva
      write(iwr,102)
      write(iwr,101) lxnv, (xnv(i,j),i=1,nnvr)
      write(iwr,101) lynv, (ynv(i,j),i=1,nnvr)
      write(iwr,101) lznv, (znv(i,j),i=1,nnvr)
50    continue
      do 52 j=1,niva
      write(iwr,102)
      write(iwr,101) lxiv, (xiv(i,j),i=1,nivr)
      write(iwr,101) lyiv, (yiv(i,j),i=1,nivr)
      write(iwr,101) lziv, (ziv(i,j),i=1,nivr)
52    continue
      return
c  *
c  ******************************************************
c  *
      entry prtvd2
c  *
c  *  PRINT THE VELOCITIES AND CLEAR THE CORRESPONDING ARRAYS
c  *
      write(iwr,105)
105   format(" *** PRTVD ***")
      idel=1
      goto 60
c  *
c  *
c  *******************************************
c  *
      entry clv2
c  *
c  *  CLEAR THE ARRAYS OF INDUCED VELOCITIES
c  *
70    continue
      do 80 i=1,nncr
      do 80 j=1,nnca
      wxnc(i,j)=0.
      wync(i,j)=0.
      wznc(i,j)=0.
80    continue
c  *
      do 81 i=1,nicr
      do 81 j=1,nica
      wxic(i,j)=0.
      wyic(i,j)=0.
      wzic(i,j)=0.
81    continue
c  *
```

```
        do 82 i=1,nncr2
        do 82 j=1,nncr2
        twx(i,j)=0.
        twy(i,j)=0.
        twz(i,j)=0.
82      continue
        do 83 i=1,nncr2
        wxctr(i)=0.
        wyctr(i)=0.
        wzctr(i)=0.
83      continue
        do 84 i=irl1,irl2
        do 85 j=1,niva
        wyivr(i,j)=0.
        wzivr(i,j)=0.
        wxivr(i,j)=0.
85      continue
84      continue
        do 5017 i1=nrl1,nrl2
        do 5017 j1=1,nnva
        wxnvr(i1,j1)=0.
        wynvr(i1,j1)=0.
        wznvr(i1,j1)=0.
5017    continue
        return
c   *
c   ***************************************************
c   *
        entry prtvn2
c   *
c   *  PRINT THE VELOCITIES AT THE CENTERS OF THE NEAR WAKE ONLY
c   *
        idel=2
        write(iwr,106)
106     format(" *** PRTVN ***")
        goto 60
c   *
c   ***************************************************
c   *
        entry prtvi2
c   *
c   *  PRINT THE VELOCITIES AT THE CENTERS OF THE INTERMEDIATE WAKE ONLY
c   *
        idel=3
        write(iwr,107)
107     format(" *** PRTVI ***")
        goto 61
c   *
        end
```

```
      subroutine vind2
c  *  *********************
c  *  *                   *
c  *  *   PROGRAM FWC     *
c  *  *                   *
c  *  *  SUBROUTINE VIND2  *
c  *  *                   *
c  *  *********************
c  *
c  *  07/14/80
c  *
c  *  LOOP ON THE POSITIONS WHERE THE INDUCED VELOCITIES
c  *  ARE COMPUTED
c  *  ENTRIES : VINDB : VELOCITIES INDUCED ON THE BLADES
c  *                    (EVALUATION OF THE INFLUENCE COEFFICIENTS)
c  *            VINDN : VELOCITIES INDUCED ON THE BLADES NEAR WAKE
c  *            VINDI : VELOCITIES INDUCED ON THE BLADES
c  *                    INTERMEDIATE WAKE
c  *            EXECN : VERIFICATION OF THE INFLUENCE COEFFICIENTS
c  *            INICN : INITIALISATION OF THE ARRAYS FOR
c  *                    VERIFICATION OF THE INFLUENCE COEFFICIENTS
c  *
      common/rol16/q1,q2,q3,q4,iw
      common/rol15/wxctr(25),wyctr(25),wzctr(25)
      common/rol11/hammt5(26),hammt6(26)
      common/rol12/wzivr(8,50),wxivr(8,50),wyivr(8,50),wznvr(28,18),
     &wxnvr(28,18),wynvr(28,18)
      common/cvind/hm1n(24,24),hm2n(24,24),hammt1(24),hammt2(24),hammt3(24)
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &ltwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nr11,nr12,ir11,ir12,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      common /vol/x,y,z,ux,uy,uz,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,
     &gm,gm1,gm2,dgm,it,rho,zof,t1,eps1,eps2,stre,jgm
      common /vort/gm1n(24),gm2n(24),gm1i(5),gm2i(5),
     &gammt1,gammt2,gammt3,gammt4
      dimension wzr(1,6)
      dimension wxcc(24),wycc(24),wzcc(24)
      dimension wxcca(24),wycca(24),wzcca(24)
      common/rollup/cg,icg,p,q
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      data lxc,lyc,lzc/"WXC ","WYC ","WZC "/
      data lxca,lyca,lzca/"WXCA","WYCA","WZCA"/
c  *
c  *
      entry vindb2(k,ikk)
```

```
c    *
c    * CALCULATION OF THE INFLUENCE COEFFICIENTS
c    *
          irl21=irl2-1
c         irl22=irl2-2
          sav1=gm
          sav2=dgm
          nbl=nblds
          if(it.eq.3) nbl=1
          if(k.eq.4) go to 101
          nncr3=nncr2-1
          do 1 j=1,nncr2
          goto (11,12,13),k
c    *
c    * NEAR WAKE, SURFACE ELEMENTS
c    *
11        continue
          if(hm1n(ikk,j).eq.0.and.hm2n(ikk,j).eq.0) goto 1
          gm=.5*(hm1n(ikk,j)+hm2n(ikk,j))
          dgm=.5*(hm2n(ikk,j)-hm1n(ikk,j))
c    *
          gm=gm*.5
          dgm=dgm*.5
          do 3 kk=1,nncr2
          do 3 jj=1,2
          x=xnc(kk+1,jj)
          y=ync(kk+1,jj)
          z=znc(kk+1,jj)
          cc=1.
          ss=0.
          do 3 n=1,nbl
          call wxyz
c    *
c    * ROTATION OF THE VELOCITY VECTOR OF -(N-1)*2.*PI/BLADES
          twx(kk,j)=twx(kk,j)+ux*cc+uy*ss
          twy(kk,j)=twy(kk,j)+uy*cc-ux*ss
          twz(kk,j)=twz(kk,j)+uz
c    *
c    * ROTATION OF THE POINT OF +2.*PI/BLADES
          sav=x
          x=sav*c-y*s
          y=s*sav+y*c
c    * INCREMENT CCSS OF 2.*PI/BLADES
          sav=cc
          cc=cc*c-ss*s
          ss=ss*c+sav*s
3         continue
c    *
          goto 1
c    *
c    * NEAR WAKE, SEGMENT ELEMENTS
c    *
12        continue
          if(ikk.eq.1.and.hammt1(j).eq.0.) goto 1
          if(ikk.eq.nnvr1.and.hammt2(j).eq.0.) goto 1
          gm=hammt1(j)
          if(ikk.eq.nnvr1)gm=hammt2(j)
          goto 20
c    *
c    *
```

```
13        continue
          if(hm11(ikk,j).eq.0.and.hm21(ikk,j).eq.0) goto 1
          gm=.5*(hm11(ikk,j)+hm21(ikk,j))
          dgm=.5*(hm21(ikk,j)-hm11(ikk,j))
          goto 20
c    *
c    *
c    *
20        continue
          do 4 kk=1,nncr2
          x=etanc(kk+1)
          y=0.
          z=0.
          cc=1.
          ss=0.
          do 4 n=1,nbl
          call wxyz
          twx(kk,j)=twx(kk,j)+ux*cc+uy*ss
          twy(kk,j)=twy(kk,j)+uy*cc-ux*ss
          twz(kk,j)=twz(kk,j)+uz
          sav=x
          x=sav*c-y*s
          y=s*sav+y*c
          sav=cc
          cc=cc*c-ss*s
          ss=ss*c+sav*s
4         continue
1         continue
          gm=sav1
          dgm=sav2
          go to 102
c    *
c *  ROLLED UP VORTICES
c    *
101       gm=q1
          if(ikk.eq.ir12) gm=q4
          if(ikk.eq.ir121) gm=q3
c         if(ikk.eq.ir122) gm=q2
          do 44 kk=1,nncr2
          x=etanc(kk+1)
          y=0.
          z=0.
          cc=1.
          ss=0.
          do 44 n=1,nbl
          call wxyz
          wxctr(kk)=wxctr(kk)+ux*cc+uy*ss
          wyctr(kk)=wyctr(kk)+uy*cc-ux*ss
          wzctr(kk)=wzctr(kk)+uz
          sav=x
          x=sav*c-y*s
          y=s*sav+y*c
          sav=cc
          cc=cc*c-ss*s
          ss=ss*c+sav*s
44        continue
          gm=sav1
          dgm=sav2
          if(itrent.eq.0)return
102       if(itrent.eq.0) return
```

```
c    *
c    ****************************************
c    *
c    *  CONTROL: VERIFICATION OF THE INFLUENCE COEFFICIENTS
c    *
          if(k.ne.1) goto 302
          do 301 kk=1,nncr2
          do 301 jj=1,2
          x=xnc(kk+1,jj)
          y=ync(kk+1,jj)
          z=znc(kk+1,jj)
          cc=1.
          ss=0.
          do 301 n=1,nbl
          call wxyz
          wxcc(kk)=wxcc(kk)+(ux*cc+uy*ss)*.5
          wycc(kk)=wycc(kk)+(uy*cc-ux*ss)*.5
          wzcc(kk)=wzcc(kk)+uz*.5
          sav=x
          x=sav*c-y*s
          y=s*sav+y*c
          sav=cc
          cc=cc*c-ss*s
          ss=ss*c+sav*s
301       continue
          return
c    *
302       continue
          do 304 kk=1,nncr2
          x=etanc(kk+1)
          y=0.
          z=0.
          cc=1.
          ss=0.
          do 304 n=1,nbl
          call wxyz
          wxcc(kk)=wxcc(kk)+(ux*cc+uy*ss)
          wycc(kk)=wycc(kk)+(uy*cc-ux*ss)
          wzcc(kk)=wzcc(kk)+uz
          sav=x
          x=sav*c-y*s
          y=s*sav+y*c
          sav=cc
          cc=cc*c-ss*s
          ss=ss*c+sav*s
304       continue
c    *
          return
c    *
c    *
c    ********************************
c    *
          entry vindn2(k1,k2)
c    *
c    *  VELOCITIES INDUCED ON THE NEAR WAKE CONTINOUS VORTEX SHEET
c    *
          nbl=nblds
          if(it.eq.3)nbl=1
          do 15 i1=1,nncr1
          do 15 j1=k1,k2
```

```
      x=xnc(i1,j1)
      y=ync(i1,j1)
      z=znc(i1,j1)
      cc=1.
      ss=0.
      do 15 n=1,nbl
      call wxyz
      wxnc(i1,j1)=wxnc(i1,j1)+ux*cc+uy*ss
      wync(i1,j1)=wync(i1,j1)+uy*cc-ux*ss
      wznc(i1,j1)=wznc(i1,j1)+uz
      sav=x
      x=sav*c-y*s
      y=s*sav+y*c
      sav=cc
      cc=cc*c-ss*s
      ss=ss*c+sav*s
15    continue
      return
c  *
c  * VELOCITIES INDUCED AT THE NODES OF THE ANTICIPATED NEAR WAKE ROLLED UP
c  * VORTICES
c  *
      entry vindn12(k1,k2)
      nbl=nblds
      if(it.eq.3) nbl=1
      ip=0
      do 5010 k3=nr11,nr12
      do 5010 j1=2,nnva
      x=xnv(k3,j1)
      y=ynv(k3,j1)
      z=znv(k3,j1)
      cc=1.
      ss=0.
      ip=0
      if(x1.eq.x.or.x3.eq.x) ip=2
      if(y1.eq.y.or.y3.eq.y) ip=2
      do 5010 n=1,nbl
      if(it.eq.3) go to 3000
      if(n.eq.1.and.ip.eq.2) go to 3001
3000  call wxyz
      if(it.eq.8.or.it.eq.9) go to 5011
      wxnvr(k3,j1)=wxnvr(k3,j1)+ux*cc+uy*ss
      wynvr(k3,j1)=wynvr(k3,j1)+uy*cc-ux*ss
5011  wznvr(k3,j1)=wznvr(k3,j1)+uz
3001  sav=x
      x=sav*c-y*s
      y=s*sav+y*c
      sav=cc
      cc=cc*c-ss*s
      ss=ss*c+sav*s
5010  continue
      return
c  *
c  * VELOCITIES INDUCED AT THE NODES OF INTERMEDIATE WAKE ROLLED UP VORTICES
c  *
      entry vindi2(k1,k2)
c  *
c  *
      nbl=nblds
      if(it.eq.3)nbl=1
```

```
      do 5020 k3=irl1,irl2
      do 5020 j=2,niva
      x=xiv(k3,j)
      y=yiv(k3,j)
      z=ziv(k3,j)
      cc=1.
      ss=0.
      ip=0
      if(x1.eq.x.or.x3.eq.x) ip=1
      do 5021 n=1,nbl
      if(it.eq.3) go to 3002
      if(n.eq.1.and.ip.eq.1) go to 3003
3002  call wxyz
      wxivr(k3,j)=wxivr(k3,j)+ux*cc+uy*ss
      wyivr(k3,j)=wyivr(k3,j)+uy*cc-ux*ss
      wzivr(k3,j)=wzivr(k3,j)+uz
3003  sav=x
      x=sav*c-y*s
      y=s*sav+y*c
      sav=cc
      cc=cc*c-ss*s
      ss=ss*c+sav*s
5021  continue
5020  continue
      return
c   *
c   ***********************************
c   *
c   *  VERIFICATION OF THE INFLUENCE COEFFICIENTS
c   *  CALCULATION OF THE VELOCITIES BY THE INFLUENCE COEFFICIENTS
c   *  AND PRINT OUT
c   *
      entry execn2
c   *
      do 404 i=1,nncr2
      wxcca(i)=0.
      wycca(i)=0.
      wzcca(i)=0.
      do 404 j=1,nncr2
      wxcca(i)=wxcca(i)+twx(i,j)*gamme(j+1)
      wycca(i)=wycca(i)+twy(i,j)*gamme(j+1)
      wzcca(i)=wzcca(i)+twz(i,j)*gamme(j+1)
404   continue
      write(iwr,100)
100   format(" *** EXECN ***")
      write(iwr,110) lxc  ,(wxcc(ki),ki=1,nncr2)
      write(iwr,110) lyc  ,(wycc(ki),ki=1,nncr2)
      write(iwr,110) lzc  ,(wzcc(ki),ki=1,nncr2)
      write(iwr,110) lxca,(wxcca(ki),ki=1,nncr2)
      write(iwr,110) lyca,(wycca(ki),ki=1,nncr2)
      write(iwr,110) lzca,(wzcca(ki),ki=1,nncr2)
      return
c   *
c   ***************************************
c   *
      entry inicn2
c   *
      do 405 i=1,nncr2
      wxcc(i)=0.
      wycc(i)=0.
```

```
      wzcc(I)=0.
405   continue
      return
c  *
110   format(1x,a4,1x,6f10.6,/,6x,6f10.6,/,6x,6f10.6)
c  *
      end
```

```
      subroutine indvel
      real*4 l0
      common /vel/x,y,z,ux,uy,uz,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,
     &gm,gm1,gm2,dgm,it,rho,zpf,tl,eps1,eps2,stre,jgm
      common /farc/eta(40),wx(22,40),wy(22,40),wz(22,40)
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/semi/dolz
c  *
      data pi/3.141592653/
c  *  fpi : value of 1./(4.*pi)
      data fpi,twopi/.079577472,6.283185306/
      data aa,bb,cc,dd/5.12160e-2,11.0501,-6.76976e-7,.527592/
c  *
c  *
      entry coord
      save zmod2,xct,yct,zct,dx,dy,xx1,yy1,zz1,xx2,yy2,zz2,xx3,yy3,zz3,
     &rmax2,ds,ds2,dsz,dsy,dsx,fva,eps12
      if(it.eq.2) go to 50
c  *
c  *   surface element
      gm=.5*(gm1+gm2)
      dgm=.5*(gm2-gm1)
      eps12=.25*eps1**2
c  *
      agm1=abs(gm1)
      agm2=abs(gm2)
      xct=(x1+x2+x3+x4)*.25
      yct=(y1+y2+y3+y4)*.25
      zct=(z1+z2+z3+z4)*.25
c  *  first unit vector
      xx1=agm1*(x3-x1)+agm2*(x4-x2)
      yy1=agm1*(y3-y1)+agm2*(y4-y2)
      zz1=agm1*(z3-z1)+agm2*(z4-z2)
      zmod1=sqrt(xx1*xx1+yy1*yy1+zz1*zz1)
      tz1=1./zmod1
      xx1=xx1*tz1
      yy1=yy1*tz1
      zz1=zz1*tz1
c  *  second unit vector
      xx2=(x2+x4-x1-x3)
      yy2=(y2+y4-y1-y3)
      zz2=(z2+z4-z1-z3)
      zmod2=sqrt(xx2*xx2+yy2*yy2+zz2*zz2)
c  *  find v2 corrected such as v1.v2=0
c  * we have v=a*v1+b*v2
c  *  /v1/=1,/v2/=1.,/v/=1, v.v2.gt.0 v.v1=0
      dot=xx1*xx2+yy1*yy2+zz1*zz2
      b=1./sqrt(zmod2*zmod2-dot*dot)
      a=-b*dot
      xx2=a*xx1+b*xx2
      yy2=a*yy1+b*yy2
      zz2=a*zz1+b*zz2
      zz3=xx1*yy2-yy1*xx2
      xx3=yy1*zz2-zz1*yy2
      yy3=zz1*xx2-xx1*zz2
c  *
c  *  dimensions of the elements
```

```
          zmod2=zmod2*.5
          dx=zmod1*.5/(agm1+agm2)
          dy=zmod2*.5
          ds=zmod1*zmod2/(agm1+agm2)
          rmax2=400.*ds
          return
c *
53        continue
          xct=.5*(x1+x3)
          yct=.5*(y1+y3)
          zct=.5*(z1+z3)
          z0=zct
          r1=sqrt(x1*x1+y1*y1)
          r3=sqrt(x3*x3+y3*y3)
          r0=.5*(r1+r3)
          dsx=.5*(x3-x1)
          dsy=.5*(y3-y1)
          dsz=.5*(z3-z1)
          ds2=dsx**2+dsy**2+dsz**2
          fvc=eps2*eps2*ds2
          rmax2=400.*ds2
          return
c *
c *
          entry wxyz
c *
c *
          ipr=6
          if(it.eq.4) go to 88
          if(it.eq.3) go to 77
          dxx=x-xct
          dyy=y-yct
          dzz=z-zct
          r02=dxx**2+dyy**2+dzz**2
          if(it.eq.2) go to 2
          if(it.eq.9) go to 2
          gc=gm/zmod2
          gv=dgm/(zmod2*dy)
          xp=xx1*dxx+yy1*dyy+zz1*dzz
          yp=xx2*dxx+yy2*dyy+zz2*dzz
          zp=xx3*dxx+yy3*dyy+zz3*dzz
          if(r02.lt.rmax2) go to 4
          r03=r02*sqrt(r02)
          f=ds*fpi*gc/r03
          wyp=-f*zp
          wzp=f*yp
          go to 10
4         continue
c *
c *
c**       correction of zp
          zpp=zp
          if(abs(zp).gt.0.1) go to 63
          if(zp) 60,61,61
60        zp=-sqrt(zp**2+eps12)
          go to 63
61        zp=sqrt(zp**2+eps12)
63        continue
c *
          x1p=xp+dx
```

```
      xlm=xp-dx
      ylp=yp+dy
      ylm=yp-dy
      xlp2=xlp*xlp
      xlm2=xlm*xlm
      ylp2=ylp*ylp
      ylm2=ylm*ylm
      zl2=zp*zp
      r1=sqrt(xlp2+ylp2+zl2)
      r2=sqrt(xlm2+ylp2+zl2)
      r3=sqrt(xlp2+ylm2+zl2)
      r4=sqrt(xlm2+ylm2+zl2)
      arg1=r2*zp*xlp*ylp-r1*zp*xlm*ylp
      arg2=r4*zp*xlp*ylm-r3*zp*xlm*ylm
      arg3=r1*r2*zl2+xlp*xlm*ylp2
      arg4=r3*r4*zl2+xlp*xlm*ylm2
      arg7=arg3*arg4
      sigf=1.
      if(arg7.lt.0.) sigf=-1.
      arg5=sigf*(arg1*arg4-arg2*arg3)
      arg6=sigf*(arg7+arg1*arg2)
      ajl=atan(arg5/arg6)
      if(arg1.gt.0..and.arg3.lt.0.) ajl=ajl+pi
      if(arg1.lt.0..and.arg3.lt.0.) ajl=ajl-pi
      if(arg2.gt.0..and.arg4.lt.0.) ajl=ajl-pi
      if(arg2.lt.0..and.arg4.lt.0.) ajl=ajl+pi
      if(arg5.gt.0..and.arg6.lt.0.) ajl=ajl+pi
      if(arg5.lt.0..and.arg6.lt.0.) ajl=ajl-pi
      aj2=alog((xlp+r1)*(xlm+r4)/((xlm+r2)*(xlp+r3)))
      kover=0
      if(abs(aj2).lt.5.03) go to 215
      aj2=sign(5.03,aj2)
      kover=kover+1
215   continue
      tpl=0.00001
      if(abs(ylp+r1).lt.tpl) go to 91
      if(abs(ylm+r3).lt.tpl) go to 91
      ajp1=xlp*alog((ylp+r1)/(ylm+r3))
      go to 93
91    ajp1=0.
93    if(abs(ylp+r2).lt.tpl) go to 94
      if(abs(ylm+r4).lt.tpl) go to 94
      ajp2=xlm*alog((ylp+r2)/(ylm+r4))
      go to 96
94    ajp2=0.
96    aj3=ajp1-ajp2
      if(kover.ne.0) write(6,226) kover,
     &x,y,z,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,xp,yp,zp,r1,r2,r3,r4,dx,
     &dy,aj2,aj3
226   format(//1x,"* control point is too close to the edge of the vortex
     &sheet * kover=",i2/5x,9f10.6/5x,9f10.6/5x,6f10.6,2e15.6)
      arg1=zp*(xlp+ylp+r1-xlm-ylp-r2)
      arg2=zp*(xlp+ylm+r3-xlm-ylm-r4)
      arg3=zl2+(xlp+ylp+r1)*(xlm+ylp+r2)
      arg4=zl2+(xlp+ylm+r3)*(xlm+ylm+r4)
      arg7=arg3*arg4
      sigf=1.
      if(arg7.lt.0.) sigf=-1.
      arg5=sigf*(arg1*arg4-arg2*arg3)
      arg6=sigf*(arg7+arg1*arg2)
```

```
          bj3=atan(arg5/arg6)
          if(arg1.gt.0..and.arg3.lt.0.) bj3=bj3+pi
          if(arg1.lt.0..and.arg3.lt.0.) bj3=bj3-pi
          if(arg2.gt.0..and.arg4.lt.0.) bj3=bj3-pi
          if(arg2.lt.0..and.arg4.lt.0.) bj3=bj3+pi
          if(arg5.gt.0..and.arg6.lt.0.) bj3=bj3+pi
          if(arg5.lt.0..and.arg6.lt.0.) bj3=bj3-pi
          aj3=aj3+2.*zp*bj3
          wyp=-fpi*(gc*aj1+gv*(yp*aj1+zpp*aj2))*(z12-eps12)/z12
          wzp=-fpi*(gc*aj2+gv*(yp*aj2+aj3))
c  *  velocities in the general system of coordinates
10        ux=wyp*xx2+wzp*xx3
          uy=wyp*yy2+wzp*yy3
          uz=wyp*zz2+wzp*zz3
          return
c  *
c  *
c  *
c  *  concetrated straight vortex element
c  *
2         continue
          dsmx=dsz*dyy-dsy*dzz
          dsmy=dsx*dzz-dsz*dxx
          dsmz=dsy*dxx-dsx*dyy
          dsm2=dsmx*dsmx+dsmy*dsmy+dsmz*dsmz
          if(dsm2.le..1e-20) go to 113
          fvds=fva/dsm2
          i0=1.
          r03=r02*sqrt(r02)
          if(r02.gt.rmax2) go to 7
c  *  zero induced velocity on vortex itself
          if(r02.lt..1e-20) go to 113
          a=-(dxx*dsx+dyy*dsy+dzz*dsz)/r02
          if(abs(a-1.).le..001) goto 113
          alpha2=ds2/r02
          alpaa=alpha2-a*a
          if(alpha2+(2.*a).lt.-1.or.alpha2-(2.*a).lt.-1.)goto 113
          if(abs(alpaa).le..1e-15) go to 334
          sq1a=sqrt(1.+2.*a+alpha2)
          sq2a=sqrt(1.-2.*a+alpha2)
          if((sq1a.le..1e-20).or.(sq2a.le..1e-20)) write(6,337)sq1a,sq2a,x,
     &y,z,x1,y1,z1,x3,y3,z3
337       format(///1x,"**overflow**",3x,"sq1a,sq2a",2e15.5/3x,"x,y,z",3e15
     &.5/3x,"x1,y1,z1,x3,y3,z3",6e15.5)
          if(sq1a.le..1e-15) sq1a=sq1a+0.0001
          if(sq2a.le..1e-15) sq2a=sq2a+0.0001
          i0=((alpha2+a)/sq1a+(alpha2-a)/sq2a)/(2.*alpaa)
          go to 7
334       i0=1./((1.-a*a)**2)
7         fact=-i0*fpi*gm*2./r03
c  *  induced velocity multiplied by 1.(1.+(eps2/rho)**2)
c  *  =1./(1.+fva/dsm2)
c  *  zero induced velocity on vortex line itself
          if(dsm2.lt..1e-20) go to 113
          fvds=fva/dsm2
          fact=fact/(1.+fvds)
          ux=fact*dsmx
          uy=fact*dsmy
          uz=fact*dsmz
          return
```

```
113       ux=0.
          uy=0.
          uz=0.
          return
c  *
c  *
c  *
c  *  semi infinite cylinder
77        continue
88                  continue
c   *
  *
          etar=sqrt(x**2+y**2)
          zp=zpf-z
          uz1=0.
          do 1000 k=1,10
          psr=.1571+(k-1)*0.3142
          u1=rho*rho+etar*etar+zp*zp-2*rho*etar*cos(psr)
          u2=rho*0.3142*(rho-etar*cos(psr))/(etar*etar+rho*rho-2*rho*etar*
         &cos(psr))*(1.-zp/(sqrt(u1)))
          uz1=(gm/12.5664)*u2+uz1
1000      continue
          uz=2*1.*uz1/delz
          sqr=(4*etar*rho)/(zp**2+(rho+etar)**2)
          sqrr=sqrt(sqr)
          sq=1.-sqr
          sqrp=sqrt(sq)
          sqrpp=sqrp**2
          if(sqr.eq.1.) go to 1001
          capk=log(4./sqrp)+0.25*(log(4./sqrp)-1.)*sqrpp
          e=1.+.5*(log(4./sqrp)-.5)*sqrpp
          go to 1002
1001      capk=10.0
          e=1.0
1002      uh=-(2./sqrr)*sqrt(rho/etar)*(capk*(2.-sqr)-2.*e)
          ps=atan2(y,x)
          ux=(uh*1.*cos(ps))*gm/(12.5664*delz)
          uy=(uh*1.*sin(ps))*gm/(12.5664*delz)
          return
          end
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      subroutine wnofc2
c   *
c   *  *********************
c   *  *                   *
c   *  *   PROGRAM FWC     *
c   *  *                   *
c   *  *  SUBROUTINE WNOFC2 *
c   *  *                   *
c   *  *********************
c   *
c   *     2/13/78
c   *
c   *  EVALUATION OF THE INDUCED VELOCITIES AT THE NODES
c   *  BY INTERPOLATION OF THE VELOCITIES INDUCED AT THE CENTERS
c   *  AND INTERPOLATION NEW/OLD DISTRIBUTION OF INDUCED VELOCITIES
c   *
c   *  ENTRY INIWN: EVALUATION OF THE INTERPOLATION COEFFICIENTS
c   *
      common/lamb/wis
      common/roll6/q1,q2,q3,q4,iw
      common/roll1/hammt5(26),hammt6(26)
      common/ali/cg1,cg2,cg3,cg4,ir
      common/roll2/wzivr(8,50),wxivr(8,50),wyivr(8,50),wznvr(28,18),
     &wxnvr(28,18),wynvr(28,18)
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &ltwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alpnas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      common /cwnof/aan(25),aai(25),icb(25),acb(25),jcb(26),bcb(26),
     &icc(7),acc(7),jcc(3),bcc(3),frn,ftn,fzn
      common /wndata/wxnvt(28,18),wynvt(28,18),wznvt(28,18)
     &,wxivt(8,50),wyivt(8,50),wzivt(8,50)
      common /savtw/twyt(24,24),twzt(24,24)
      common/tipco/wxnt1,wynt1,wznt1,wxnt2,wynt2,wznt2,wxntp,wyntp,wzntp
      dimension wrnvt(28,44),wtnvt(28,44),wrivt(8,50),wtivt(8,50)
      dimension rs(28,50)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      equivalence  (wxnvt(1,1),wrnvt(1,1)),(wynvt(1,1),wtnvt(1,1))
      equivalence  (wxivt(1,1),wrivt(1,1)),(wyivt(1,1),wtivt(1,1))
      data laan,laai,licb,lacb,ljcb/" AAN"," AAI"," ICB"," ACB"," JCB"/
      data lbcb,licc,lacc,ljcc,lbcc/" BCB"," ICC"," ACC"," JCC"," BCC"/
      data lxnv,lynv,lznv,lrnv,ltnv/"WXNV","WYNV","WZNV","WRNV","WTNV"/
      data lxiv,lyiv,lziv,lriv,ltiv/"WXIV","WYIV","WZIV","WRIV","WTIV"/
      data ltwy,ltwz/"TWY ","TWZ "/
      data dps2,fpi/100.,12.566371/
c   *
```

```
      goto 70
c   *
c   ****************************************
c   *
      entry iniwn2
c   *
      if(isamet.eq.1) return
c   *
c   * INTERPOLATION COEFFICIENTS
c   *
c   * COEFFICIENTS AAN : NODE I VS. CENTERS I AND I+1
c   * FOR THE NEAR WAKE, RADIAL DIRECTION
      do 50 i=1,nnvr
      aan(i)=(etanc(i+1)-etanv(i))/(etanc(i+1)-etanc(i))
50    continue
c   *
      do 51 i=1,nivr
      aai(i)=(etaic(i+1)-etaiv(i))/(etaic(i+1)-etaic(i))
51    continue
c   *
c   * COEFFICIENTS ACB AND THE CORRESPONDING ADRESSES ICB :
c   * NODE I OF THE NEAR WAKE VS. THE CENTERS ICB(I) AND ICB(I)+1,
c   * OF THE INT. WAKE (RADIAL DIRECTION)
      do 52 i=1,nnvr
      et=etanv(i)
      do 53 i1=2,nicr
      if(et.lt.etaic(i1)) goto 54
53    continue
54    icb(i)=i1-1
      acb(i)=(etaic(i1)-et)/(etaic(i1)-etaic(i1-1))
52    continue
c   *
c   * COEFFICIENTS BCB AND THE CORRESPONDING ADRESSES JCB :
c   * NODE J OF THE TRANSITION WAKE VS. CENTERS JCB(J) AND JCB(J)+1
c   * OF THE INTERMEDIATE WAKE, AZIMUTAL DIRECTION
      do 55 j=ntca,nnva
      et=(j-1)*dpsin
      do 56 j1=2,4
      eti=(ntva-1)*dpsin+(j1-1)*dpsii-dpsii*.5
      if(et.lt.eti) goto 57
56    continue
57    jcb(j)=j1-1
      bcb(j)=(eti-et)/dpsii
55    continue
c   *
c   *
c   * COEFFICIENTS BCC AND THE CORRESPONDING ADRESSES JCC :
c   * NODES J OF THE INT. WAKE VS. THE NODES JCC(J) AND JCC(J)+1
c   * OF THE NEAR WAKE (AZIMUTAL DIRECTION )
      do 58 i=1,nivr
      et=etaiv(i)
      do 59 i1=2,nnvr
      if(et.lt.etanv(i1)) go to 60
59    continue
60    icc(i)=i1-1
      acc(i)=(etanv(i1)-et)/(etanv(i1)-etanv(i1-1))
58    continue
c   *
      do 61 j=1,3
      et=(ntva-1)*dpsin+(j-1)*dpsii
```

```
      do 62 j1=ntca,nnva
      eti=(j1-1)*dpsin
      if(j.eq.1.and.j1.eq.ntca.or.et.lt.eti) goto 63
62    continue
63    jcc(j)=j1-1
      bcc(j)=(eti-et)/dpsin
61    continue
c
      if(itrace.eq.0) return
      write(iwr,200)
200   format(" *** INIWNOFC ***")
      write(iwr,101) laan,(aan(i),i=1,nnvr)
      write(iwr,101) laai,(aai(i),i=1,nivr)
      write(iwr,102) licb,(icb(i),i=1,nnvr)
      write(iwr,101) lacb,(acb(i),i=1,nnvr)
      write(iwr,102) ljcb,(jcb(i),i=ntca,nnva)
      write(iwr,101) lbcb,(bcb(i),i=ntca,nnva)
      write(iwr,102) licc,(icc(i),i=1,nivr)
      write(iwr,101) lacc,(acc(i),i=1,nivr)
      write(iwr,102) ljcc,(jcc(i),i=1,3)
      write(iwr,101) lbcc,(bcc(i),i=1,3)
      return
c   *
c   *****************************************************
c   *
70    continue
c   *
c   *  INTERPOLATION OF THE VELOCITIES
c   *
c   *  A) INTERPOLATION OF THE VELOCITIES INDUCED AT THE CENTERS
c   *  OF THE NEAR WAKE 1 TO NNCA TO THE NODES 1 TO NNVA
      do 1 i=1,nnvr1
      do 1 j=1,nnva
      a=aan(i)
      a1=1.-a
      wxnvt(i,j)=((wxnc(i,j)+wxnc(i,j+1))*a+(wxnc(i+1,j)+
     &wxnc(i+1,j+1))*a1)*.5
      wynvt(i,j)=((wync(i,j)+wync(i,j+1))*a+(wync(i+1,j)+
     &wync(i+1,j+1))*a1)*.5
      wznvt(i,j)=((wznc(i,j)+wznc(i,j+1))*a+(wznc(i+1,j)+
     &wznc(i+1,j+1))*a1)*.5
1     continue
      a=aan(nnvr1)
      a1=1.-a
      wxnvt(nnvr1,1)=wxntp+(wxnc(nnvr1,2)-wxnt1)*a+(wxnc(nncr1,2)-wxnt2)*a1
      wynvt(nnvr1,1)=wyntp+(wync(nnvr1,2)-wynt1)*a+(wync(nncr1,2)-wynt2)*a1
      wznvt(nnvr1,1)=wzntp+(wznc(nnvr1,2)-wznt1)*a+(wznc(nncr1,2)-wznt2)*a1
      if(itrcw.ne.2) goto 10
      write(iwr,179)
179   format("1*** WNOFC *** VELOCITIES AT THE NODES OF THE NEAR WAKE")
      do 11 j=1,nnva
      write(iwr,103)
      write(iwr,101) lxnv,(wxnvt(i,j),i=1,nnvr)
      write(iwr,101) lynv,(wynvt(i,j),i=1,nnvr)
      write(iwr,101) lznv,(wznvt(i,j),i=1,nnvr)
11    continue
c   *
c   *  B) INTERPOLATION OF THE VELOCITIES AT THE CENTERS 1-2-3-4
c   *  (AZIMUTAL POSITIONS OF THE INT. WAKE) AT THE NODES OF THE NEAR WA
c   *  FROM NTCA+1 TO NNCA
```

```
10        if(itrans.eq.0) goto 12
          do 2 i=1,nnvr
          i1=icb(i)
          a=acb(i)
          a1=1.-a
          do 2 j=ntca,nnva
          j1=jcb(j)
          b=bcb(j)
          b1=1.-b
          if(j.eq.ntca) b=0.5
          wxnvt(i,j)=wxnvt(i,j)+((
           &wxic(i1,j1)*b+wxic(i1,j1+1)*b1)*a+(
           &wxic(i1+1,j1)*b+wxic(i1+1,j1+1)*b1)*a1)
          wynvt(i,j)=wynvt(i,j)+((
           &wyic(i1,j1)*b+wyic(i1,j1+1)*b1)*a+(
           &wyic(i1+1,j1)*b+wyic(i1+1,j1+1)*b1)*a1)
          wznvt(i,j)=wznvt(i,j)+((
           &wzic(i1,j1)*b+wzic(i1,j1+1)*b1)*a+(
           &wzic(i1+1,j1)*b+wzic(i1+1,j1+1)*b1)*a1)
2         continue
          if(itrcw.ne.2) goto 12
          write(iwr,178)
178       format("INODES NTCA,NNVA, AFTER ADDITION OF THE VELOCITIES",
           &" INDUCED ON CENTERS 1 TO 4 OF THE INT.WG")
          do 13 j=ntca,nnva
          write(iwr,103)
          write(iwr,101) lxnv,(wxnvt(i,j),i=1,nnvr)
          write(iwr,101) lynv,(wynvt(i,j),i=1,nnvr)
          write(iwr,101) lznv,(wznvt(i,j),i=1,nnvr)
13        continue
c    *
12        ideb=5
c
c         TOTAL INDUCED VELOCITY AT INTERMEDIATE WAKE
c
          if(itrans.eq.0) ideb=1
          nr121=nr12-1
c         nr122=nr12-2
          ir121=ir12-1
c         ir122=ir12-2
          do 1001 i=ir11,ir12
          do 2000 j=2,niva
          rs(i,j)=sqrt(xiv(i,j)**2+yiv(i,j)**2)
          if(i.eq.ir11) gmi=q1
c         if(i.eq.ir122) gmi=q2
          if(i.eq.ir121) gmi=q3
          if(i.eq.ir12) gmi=q4
          wzivt(i,j)=wzivr(i,j)+(gmi*wis)/(4*3.14159*rs(i,j))
          wxivt(i,j)=wxivr(i,j)
          wyivt(i,j)=wyivr(i,j)
2000      continue
1001      continue
c
c         TOTAL INDUCED VELOCITY AT NEAR WAKE
c
          do 5016 i=nr11,nr12
          do 5016 j=2,nnva
          rs(i,j)=sqrt(xnv(i,j)**2+ynv(i,j)**2)
          if(i.eq.nr11) gmi=q1
c         if(i.eq.nr122) gmi=q2
```

C 2

```
          if(i.eq.nr121) gmi=g3
          if(i.eq.nr12) gmi=g4
          wznvt(i,j)=wznvr(i,j)+(gmi*wis)/(4*3.14159*rs(i,j))
          wxnvt(i,j)=wxnvr(i,j)
          wynvt(i,j)=wynvr(i,j)
5016      continue
c   *
          do 111 i=2,nncr1
          if(etanc(i).gt.cg1) go to 222
 111      ibo1=i
 222      po11=cg1-etanc(ibo1)
          ibo11=ibo1-1
          po12=etanc(ibo1+1)-etanc(ibo1)
          wznvt(nr11,1)=wzc(ibo11)+(wzc(ibo11+1)-wzc(ibo11))*(po11/po12)
          wynvt(nr11,1)=wyc(ibo11)+(wyc(ibo11+1)-wyc(ibo11))*(po11/po12)
          wxnvt(nr11,1)=wxc(ibo11)+(wxc(ibo11+1)-wxc(ibo11))*(po11/po12)

c
c         do 333 i=2,nncr1
c         if(etanc(i).gt.cg2) go to 444
c 333     ibo2=i
c 444     po13=(cg2-etanc(ibo2))/(etanc(ibo2+1)-etanc(ibo2))
c         ibo21=ibo2-1
c         wznvt(nr122,1)=wzc(ibo21)+(wzc(ibo21+1)-wzc(ibo21))*po13
c         wynvt(nr122,1)=wyc(ibo21)+(wyc(ibo21+1)-wyc(ibo21))*po13
c         wxnvt(nr122,1)=wxc(ibo21)+(wxc(ibo21+1)-wxc(ibo21))*po13
c
          do 555 i=2,nncr1
          if(etanc(i).gt.cg3) go to 666
 555      ibo3=i
 666      po14=(cg3-etanc(ibo3))/(etanc(ibo3+1)-etanc(ibo3))
          ibo31=ibo3-1
          wznvt(nr121,1)=wzc(ibo31)+(wzc(ibo31+1)-wzc(ibo31))*po14
          wynvt(nr121,1)=wyc(ibo31)+(wyc(ibo31+1)-wyc(ibo31))*po14
          wxnvt(nr121,1)=wxc(ibo31)+(wxc(ibo31+1)-wxc(ibo31))*po14
c
          do 777 i =2,nncr1
          if(etanc(i).gt.cg4) go to 888
 777      ibo4=i
 888      po15=(cg4-etanc(ibo4))/(etanc(ibo4+1)-etanc(ibo4))
          ibo41=ibo4-1
          wznvt(nr12,1)=wzc(ibo41)+(wzc(ibo41+1)-wzc(ibo41))*po15
          wynvt(nr12,1)=wyc(ibo41)+(wyc(ibo41+1)-wyc(ibo41))*po15
          wxnvt(nr12,1)=wxc(ibo41)+(wxc(ibo41+1)-wxc(ibo41))*po15
c    *********************************************
c    *
c    *  TRANSFORMATION : RECTANGULAR TO POLAR COORDINATES
c    *  OF THE VELOCITIES INDUCED AT THE NODES
c    *
16        continue
          if(itrcw.eq.2) write(iwr,460)
460       format("1*** WNOFC ***: VELOCITIES AT THE NODES, POLAR COOR.")
          do 45 j=1,nnva
          do 80 i=nr11,nr12
          wxx=wxnvt(i,j)
          wyy=wynvt(i,j)
          xx=xnv(i,j)
          yy=ynv(i,j)
          r=sqrt(xx*xx+yy*yy)
          wrnvt(i,j)=(xx*wxx+yy*wyy)/r
```

```
      wtnvt(i,j)=(-wxx*yy+wyy*xx)/r
80    continue
      if(itrcw.ne.2) goto 45
      write(iwr,103)
      write(iwr,101) lrnv,(wrnvt(i,j),i=1,nnvr)
      write(iwr,101) ltnv,(wtnvt(i,j),i=1,nnvr)
45    continue
      do 46 j=2,niva
      do 81 i=ir11,ir12
      wxx=wxivt(i,j)
      wyy=wyivt(i,j)
      xx=xiv(i,j)
      yy=yiv(i,j)
      r=sqrt(xx*xx+yy*yy)
      wrivt(i,j)=(xx*wxx+yy*wyy)/r
      wtivt(i,j)=(-wxx*yy+wyy*xx)/r
81    continue
      if(itrcw.ne.2) goto 46
      write(iwr,103)
      write(iwr,101) lriv,(wrivt(i,j),i=ir11,ir12)
      write(iwr,101) ltiv,(wtivt(i,j),i=ir11,ir12)
46    continue
      wrivt(ir11,1)=wrnvt(nr11,nnva)
      wtivt(ir11,1)=wtnvt(nr11,nnva)
      wrivt(ir12,1)=wrnvt(nr12,nnva)
      wtivt(ir12,1)=wtnvt(nr12,nnva)
      wzivt(ir12,1)=wznvt(nr12,nnva)
      wzivt(ir11,1)=wznvt(nr11,nnva)
      wrivt(ir121,1)=wrnvt(nr121,nnva)
      wtivt(ir121,1)=wtnvt(nr121,nnva)
      wzivt(ir121,1)=wznvt(nr121,nnva)
c     wrivt(ir122,1)=wrnvt(nr122,nnva)
c     wtivt(ir122,1)=wtnvt(nr122,nnva)
c     wzivt(ir122,1)=wznvt(nr122,nnva)
c   *
c   ***********************************************
c   *
c   *  INTERPOLATION OF THE 'OLD' AND THE 'NEW' DISTRIBUTIONS
c   *  OF VELOCITIES
c   *
      if(itrcw.ne.0) write(iwr,462)
462   format("1*** WNOFC *** : VELOCITIES AT THE NODES ")
c   *
c   *  WEIGHTING FACTORS FOR THE 'NEW' AND THE 'OLD' VELOCITIES
      frn=.4
      ftn=.7
      fzn=.3
      if(niter.ge.3) goto 49
      frn=.5
      ftn=.8
      fzn=.5
      if(niter.eq.2) goto 49
      frn=.5
      ftn=.9
      fzn=.5
49    continue
      if(itrcw.ne.0) write(iwr,951) frn,ftn,fzn
951   format(' *** WEIGHTING FACTORS: FRN,FTN,FZN ",3f6.2)
      frv=1.-frn
```

```
           ftv=1.-ftn
           fzv=1.-fzn
c    *
           do 47 j=1,nnva
           do 43 i=nrl1,nrl2
           wrnv(i,j)=wrnvt(i,j)*frn+wrnv(i,j)*frv
           wtnv(i,j)=wtnvt(i,j)*ftn+wtnv(i,j)*ftv
           wznv(i,j)=wznvt(i,j)*fzn+wznv(i,j)*fzv
           go to 43
648        wrnv(i,j)=wrnvt(i,j)
           wtnv(i,j)=wtnvt(i,j)
           wznv(i,j)=wznvt(i,j)
43         continue
           if(itrcw.eq.0) goto 47
           write(iwr,103)
           write(iwr,101)lrnv,(wrnv(i,j),i=1,nnvr)
           write(iwr,101)ltnv,(wtnv(i,j),i=1,nnvr)
           write(iwr,101)lznv,(wznv(i,j),i=1,nnvr)
47         continue
           do 48 j=1,niva
           do 44 i=irl1,irl2
           wriv(i,j)=wrivt(i,j)*frn+wriv(i,j)*frv
           wtiv(i,j)=wtivt(i,j)*ftn+wtiv(i,j)*ftv
           wziv(i,j)=wzivt(i,j)*fzn+wziv(i,j)*fzv
           go to 44
649        wriv(i,j)=wrivt(i,j)
           wtiv(i,j)=wtivt(i,j)
           wziv(i,j)=wzivt(i,j)
44         continue
           if(itrcw.eq.0) goto 48
           write(iwr,103)
           write(iwr,101)lriv,(wriv(i,j),i=irl1,irl2)
           write(iwr,101)ltiv,(wtiv(i,j),i=irl1,irl2)
           write(iwr,101)lziv,(wziv(i,j),i=irl1,irl2)
48         continue
c    *
c    ****************************************************
c    *
c    *  INTERPOLATION OF THE INFLUENCE COEFFICIENTS
c    *
           if(niter.eq.1) return
           fzn=.5
           fzv=1.-fzn
           if(itrcw.eq.0) goto 90
           write(iwr,500)
500        format(" *** WNOFC: INTERPOLATION OF INF. COEFF ",/,
            &" INF. COEFF FROM LOOP2&")
           do 91 j=1,nncr2
           write(iwr,101)ltwy,(twy(i,j),i=1,nncr2)
           write(iwr,101)ltwz,(twz(i,j),i=1,nncr2)
           write(iwr,103)
91         continue
           write(iwr,501)
501        format(" OLD INF. COEFF. ")
           do 92 j=1,nncr2
           write(iwr,101)ltwy,(twyt(i,j),i=1,nncr2)
           write(iwr,101)ltwz,(twzt(i,j),i=1,nncr2)
           write(iwr,103)
92         continue
90         continue
```

```
c    *
          do 82 i=1,nncr2
          do 82 j=1,nncr2
          twy(i,j)=fzn*twy(i,j)+fzv*twyt(i,j)
          twz(i,j)=fzn*twz(i,j)+fzv*twzt(i,j)
82        continue
c    *
          if(itrcw.eq.0) return
          write(iwr,502)fzn,fzv
502       format(" INT. INF. COEFF.: FZN=",e20.5," FZV=",e20.5)
          do 93 j=1,nncr2
          write(iwr,101)ltwy,(twy(i,j),i=1,nncr2)
          write(iwr,101)ltwz,(twz(i,j),i=1,nncr2)
          write(iwr,103)
93        continue
          return
c    *
101       format(1x,a4,1x,5f13.9,/,6x,5f13.9,/,6x,5f13.9)
102       format(1x,a4,1x,9i13,/,6x,9i13)
103       format(1x)
c    *
          end
```

```
      subroutine wnofc3
c   *
c   *  **********************
c   *  *                    *
c   *  *   PROGRAM FWC      *
c   *  *                    *
c   *  *  SUBROUTINE WNOFC3  *
c   *  *                    *
c   *  **********************
c   *
c   *     2/13/78
c   *
c   *  EVALUATION OF THE INDUCED VELOCITIES AT THE NODES
c   *  BY INTERPOLATION OF THE VELOCITIES INDUCED AT THE CENTERS
c   *  AND INTERPOLATION NEW/OLD DISTRIBUTION OF INDUCED VELOCITIES
c   *
c   *  ENTRY INIWN: EVALUATION OF THE INTERPOLATION COEFFICIENTS
c   *
      common/lamb/wis
      common/ali/cg1,cg2,cg3,cg4,ir
      common/roll6/q1,q2,q3,q4,iw
      common/roll1/hammt5(26),hammt6(26)
      common/roll2/wzivr(8,50),wxivr(8,50),wyivr(8,50),wznvr(28,18),
     &wxnvr(28,18),wynvr(28,18)
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nr11,nr12,ir11,ir12,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      common /cwnof/aan(25),aai(25),icb(25),acb(25),jcb(26),bcb(26),
     &icc(7),acc(7),jcc(3),bcc(3),frn,ftn,fzn
      common /wndata/wxnvt(28,18),wynvt(28,18),wznvt(28,18)
     &,wxivt(8,50),wyivt(8,50),wzivt(8,50)
      common /savtw/twyt(24,24),twzt(24,24)
      common/tipco/wxnt1,wynt1,wznt1,wxnt2,wynt2,wznt2,wxntp,wyntp,wzntp
      dimension wrnvt(28,44),wtnvt(28,44),wrivt(8,50),wtivt(8,50)
      dimension rs(28,50)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      equivalence  (wxnvt(1,1),wrnvt(1,1)),(wynvt(1,1),wtnvt(1,1))
      equivalence  (wxivt(1,1),wrivt(1,1)),(wyivt(1,1),wtivt(1,1))
      data laan,laai,licb,lacb,ljcb/" AAN"," AAI"," ICB"," ACB"," JCB"/
      data lbcb,licc,lacc,ljcc,lbcc/" BCB"," ICC"," ACC"," JCC"," BCC"/
      data lxnv,lynv,lznv,lrnv,ltnv/"WXNV","WYNV","WZNV","WRNV","WTNV"/
      data lxiv,lyiv,lziv,lriv,ltiv/"WXIV","WYIV","WZIV","WRIV","WTIV"/
      data ltwy,ltwz/"TWY ","TWZ "/
      data dps2,fpi/100.,12.566371/
c   *
```

```
        goto 70
c   *
c   ***************************************
c   *
        entry iniwn3
c   *
        if(isamet.eq.1) return
c   *
c   *  INTERPOLATION COEFFICIENTS
c   *
c   *  COEFFICIENTS AAN : NODE I VS. CENTERS I AND I+1
c   *  FOR THE NEAR WAKE, RADIAL DIRECTION
        do 50 i=1,nnvr
        aan(i)=(etanc(i+1)-etanv(i))/(etanc(i+1)-etanc(i))
50      continue
c   *
        do 51 i=1,nivr
        aai(i)=(etaic(i+1)-etaiv(i))/(etaic(i+1)-etaic(i))
51      continue
c   *
c   *  COEFFICIENTS ACB AND THE CORRESPONDING ADRESSES ICB :
c   *  NODE I OF THE NEAR WAKE VS. THE CENTERS ICB(I) AND ICB(I)+1,
c   *  OF THE INT. WAKE (RADIAL DIRECTION)
        do 52 i=1,nnvr
        et=etanv(i)
        do 53 i1=2,nicr
        if(et.lt.etaic(i1)) goto 54
53      continue
54      icb(i)=i1-1
        acb(i)=(etaic(i1)-et)/(etaic(i1)-etaic(i1-1))
52      continue
c   *
c   *  COEFFICIENTS BCB AND THE CORRESPONDING ADRESSES JCB :
c   *  NODE J OF THE TRANSITION WAKE VS. CENTERS JCB(J) AND JCB(J)+1
c   *  OF THE INTERMEDIATE WAKE, AZIMUTAL DIRECTION
        do 55 j=ntca,nnva
        et=(j-1)*dpsin
        do 56 j1=2,4
        eti=(ntva-1)*dpsin+(j1-1)*dpsii-dpsii*.5
        if(et.lt.eti) goto 57
56      continue
57      jcb(j)=j1-1
        bcb(j)=(eti-et)/dpsii
55      continue
c   *
c   *
c   *  COEFFICIENTS BCC AND THE CORRESPONDING ADRESSES JCC :
c   *  NODES J OF THE INT. WAKE VS. THE NODES JCC(J) AND JCC(J)+1
c   *  OF THE NEAR WAKE (AZIMUTAL DIRECTION )
        do 58 i=1,nivr
        et=etaiv(i)
        do 59 i1=2,nnvr
        if(et.lt.etanv(i1)) go to 60
59      continue
60      icc(i)=i1-1
        acc(i)=(etanv(i1)-et)/(etanv(i1)-etanv(i1-1))
58      continue
c   *
        do 61 j=1,3
        et=(ntva-1)*dpsin+(j-1)*dpsii
```

```
      do 62 j1=ntca,nnva
      eti=(j1-1)*dpsin
      if(j.eq.1.and.j1.eq.ntca.or.et.lt.eti) goto 63
62    continue
63    jcc(j)=j1-1
      bcc(j)=(eti-et)/dpsin
61    continue
c  *
      if(itrace.eq.0) return
      write(iwr,200)
200   format(" *** INIWNOFC ***")
      write(iwr,101) laan,(aan(i),i=1,nnvr)
      write(iwr,101) laai,(aai(i),i=1,nivr)
      write(iwr,102) licb,(icb(i),i=1,nnvr)
      write(iwr,101) lacb,(acb(i),i=1,nnvr)
      write(iwr,102) ljcb,(jcb(i),i=ntca,nnva)
      write(iwr,101) lbcb,(bcb(i),i=ntca,nnva)
      write(iwr,102) licc,(icc(i),i=1,nivr)
      write(iwr,101) lacc,(acc(i),i=1,nivr)
      write(iwr,102) ljcc,(jcc(i),i=1,3)
      write(iwr,101) lbcc,(bcc(i),i=1,3)
      return
c  *
c  ************************************************************
c  *
70        continue
c  *
c  *  INTERPOLATION OF THE VELOCITIES
c  *
c  *  A) INTERPOLATION OF THE VELOCITIES INDUCED AT THE CENTERS
c  *  OF THE NEAR WAKE 1 TO NNCA TO THE NODES 1 TO NNVA
      do 1 i=1,nnvr1
      do 1 j=1,nnva
      a=aan(i)
      a1=1.-a
      wxnvt(i,j)=((wxnc(i,j)+wxnc(i,j+1))*a+(wxnc(i+1,j)+
     &wxnc(i+1,j+1))*a1)*.5
      wynvt(i,j)=((wync(i,j)+wync(i,j+1))*a+(wync(i+1,j)+
     &wync(i+1,j+1))*a1)*.5
      wznvt(i,j)=((wznc(i,j)+wznc(i,j+1))*a+(wznc(i+1,j)+
     &wznc(i+1,j+1))*a1)*.5
1     continue
      a=aan(nnvr1)
      a1=1.-a
      wxnvt(nnvr1,1)=wxntp+(wxnc(nnvr1,2)-wxnt1)*a+(wxnc(nncr1,2)-wxnt2)*a1
      wynvt(nnvr1,1)=wyntp+(wync(nnvr1,2)-wynt1)*a+(wync(nncr1,2)-wynt2)*a1
      wznvt(nnvr1,1)=wzntp+(wznc(nnvr1,2)-wznt1)*a+(wznc(nncr1,2)-wznt2)*a1
      if(itrcw.ne.2) goto 10
      write(iwr,179)
179   format("1*** WNOFC *** VELOCITIES AT THE NODES OF THE NEAR WAKE")
      do 11 j=1,nnva
      write(iwr,103)
      write(iwr,101) lxnv,(wxnvt(i,j),i=1,nnvr)
      write(iwr,101) lynv,(wynvt(i,j),i=1,nnvr)
      write(iwr,101) lznv,(wznvt(i,j),i=1,nnvr)
11    continue
c  *
c  *  B) INTERPOLATION OF THE VELOCITIES AT THE CENTERS 1-2-3-4
c  *  (AZIMUTAL POSITIONS OF THE INT. WAKE) AT THE NODES OF THE NEAR WA
c  *  FROM NTCA+1 TO NNCA
```

ORIGINAL PAGE IS OF POOR QUALITY

```
10      if(itrans.eq.0) goto 12
        do 2 i=1,nnvr
        i1=icb(i)
        a=acb(i)
        a1=1.-a
        do 2 j=ntca,nnva
        j1=jcb(j)
        b=bcb(j)
        b1=1.-b
        if(j.eq.ntca) b=0.5
        wxnvt(i,j)=wxnvt(i,j)+((
         &wxic(i1,j1)*b+wxic(i1,j1+1)*b1)*a+(
         &wxic(i1+1,j1)*b+wxic(i1+1,j1+1)*b1)*a1)
        wynvt(i,j)=wynvt(i,j)+((
         &wyic(i1,j1)*b+wyic(i1,j1+1)*b1)*a+(
         &wyic(i1+1,j1)*b+wyic(i1+1,j1+1)*b1)*a1)
        wznvt(i,j)=wznvt(i,j)+((
         &wzic(i1,j1)*b+wzic(i1,j1+1)*b1)*a+(
         &wzic(i1+1,j1)*b+wzic(i1+1,j1+1)*b1)*a1)
2       continue
        if(itrcw.ne.2) goto 12
        write(iwr,178)
178     format("1NODES NTCA,NNVA, AFTER ADDITION OF THE VELOCITIES",
         &" INDUCED ON CENTERS 1 TO 4 OF THE INT.W6")
        do 13 j=ntca,nnva
        write(iwr,103)
        write(iwr,101) lxnv,(wxnvt(i,j),i=1,nnvr)
        write(iwr,101) lynv,(wynvt(i,j),i=1,nnvr)
        write(iwr,101) lznv,(wznvt(i,j),i=1,nnvr)
13      continue
c   *
12      ideb=5
c
c       TOTAL INDUCED VELOCITY AT INTERMEDIATE WAKE
c
        if(itrans.eq.0) ideb=1
        nr121=nr12-1
c       nr122=nr12-2
        ir121=ir12-1
c       ir122=ir12-2
        do 1001 i=ir11,ir12
        do 2000 j=2,niva
        rs(i,j)=sqrt(xiv(i,j)**2+yiv(i,j)**2)
        if(i.eq.ir11) gmi=q1
c       if(i.eq.ir122) gmi=q2
        if(i.eq.ir121) gmi=q3
        if(i.eq.ir12) gmi=q4
        wzivt(i,j)=wzivr(i,j)+(gmi*wis)/(4*3.14159*rs(i,j))
        wxivt(i,j)=wxivr(i,j)
        wyivt(i,j)=wyivr(i,j)
2000    continue
1001    continue
c
c       TOTAL INDUCED VELOCITY AT NEAR WAKE
c
        do 5016 i=nr11,nr12
        do 5016 j=2,nnva
        rs(i,j)=sqrt(xnv(i,j)**2+ynv(i,j)**2)
        if(i.eq.nr11) gmi=q1
c       if(i.eq.nr122) gmi=q2
```

```
          if(i.eq.nr121) gmi=q3
          if(i.eq.nr12) gmi=q4
          wznvt(i,j)=wznvr(i,j)+(gmi*wis)/(4*3.14159*rs(i,j))
          wxnvt(i,j)=wxnvr(i,j)
          wynvt(i,j)=wynvr(i,j)
5016      continue
c    *
          do 111 i=2,nncr1
          if(etanc(i).gt.cg1) go to 222
 111      ibo1=i
 222      pol1=(cg1-etanc(ibo1))/(etanc(ibo1+1)-etanc(ibo1))
          ibo11=ibo1-1
          wznvt(nr11,1)=wzc(ibo11)+(wzc(ibo11+1)-wzc(ibo11))*pol1
          wynvt(nr11,1)=wyc(ibo11)+(wyc(ibo11+1)-wyc(ibo11))*pol1
          wxnvt(nr11,1)=wxc(ibo11)+(wxc(ibo11+1)-wxc(ibo11))*pol1
c
c         do 333 i=2,nncr1
c         if(etanc(i).gt.cg2) go to 444
c 333     ibo2=i
c 444     pol3=(cg2-etanc(ibo2))/(etanc(ibo2+1)-etanc(ibo2))
c         ibo21=ibo2-1
c         wznvt(nr122,1)=wzc(ibo21)+(wzc(ibo21+1)-wzc(ibo21))*pol3
c         wynvt(nr122,1)=wyc(ibo21)+(wyc(ibo21+1)-wyc(ibo21))*pol3
c         wxnvt(nr122,1)=wxc(ibo21)+(wxc(ibo21+1)-wxc(ibo21))*pol3
c
          do 555 i=2,nncr1
          if(etanc(i).gt.cg3) go to 666
 555      ibo3=i
 666      pol4=(cg3-etanc(ibo3))/(etanc(ibo3+1)-etanc(ibo3))
          ibo31=ibo3-1
          wznvt(nr121,1)=wzc(ibo31)+(wzc(ibo31+1)-wzc(ibo31))*pol4
          wynvt(nr121,1)=wyc(ibo31)+(wyc(ibo31+1)-wyc(ibo31))*pol4
          wxnvt(nr121,1)=wxc(ibo31)+(wxc(ibo31+1)-wxc(ibo31))*pol4
c
          do 777 i=2,nncr1
          if(etanc(i).gt.cg4) go to 888
 777      ibo4=i
 888      pol5=(cg4-etanc(ibo4))/(etanc(ibo4+1)-etanc(ibo4))
          ibo41=ibo4-1
          wznvt(nr12,1)=wzc(ibo41)+(wzc(ibo41+1)-wzc(ibo41))*pol5
          wynvt(nr12,1)=wyc(ibo41)+(wyc(ibo41+1)-wyc(ibo41))*pol5
          wxnvt(nr12,1)=wxc(ibo41)+(wzc(ibo41+1)-wzc(ibo41))*pol5
c    ********************************************
c    *
c    *  TRANSFORMATION : RECTANGULAR TO POLAR COORDINATES
c    *  OF THE VELOCITIES INDUCED AT THE NODES
c    *
16        continue
          if(itrcw.eq.2) write(iwr,460)
460       format("1*** WNOFC ***: VELOCITIES AT THE NODES, POLAR COOR.")
          do 45 j=1,nnva
          i1=nr11
          i2=nr12
 7000     do 80 i=i1,i2
          if(i.eq.nr12) i3=0
          if(i.ne.nr12) i3=1
          wxx=wxnvt(i,j)
          wyy=wynvt(i,j)
          xx=xnv(i,j)
          yy=ynv(i,j)
```

```
          r=sqrt(xx*xx+yy*yy)
          wrnvt(i,j)=(xx*wxx+yy*wyy)/r
          wtnvt(i,j)=(-wxx*yy+wyy*xx)/r
80        continue
          if(itrcw.ne.2) goto 7001
          write(iwr,103)
          write(iwr,101)lrnv,(wrnvt(i,j),i=1,nnvr)
          write(iwr,101)ltnv,(wtnvt(i,j),i=1,nnvr)
 7001     i1=1
          i2=nnvr
          if(i3.eq.0) go to 7000
45        continue
          do 46 j=2,niva
          do 81 i=irl1,irl2

          wxx=wxivt(i,j)
          wyy=wyivt(i,j)
          xx=xiv(i,j)
          yy=yiv(i,j)
          r=sqrt(xx*xx+yy*yy)
          wrivt(i,j)=(xx*wxx+yy*wyy)/r
          wtivt(i,j)=(-wxx*yy+wyy*xx)/r
81        continue
          if(itrcw.ne.2) goto 46
          write(iwr,103)
          write(iwr,101)lriv,(wrivt(i,j),i=irl1,irl2)
          write(iwr,101)ltiv,(wtivt(i,j),i=irl1,irl2)
46        continue
          wrivt(irl1,1)=wrnvt(nrl1,nnva)
          wtivt(irl1,1)=wtnvt(nrl1,nnva)
          wrivt(irl2,1)=wrnvt(nrl2,nnva)
          wtivt(irl2,1)=wtnvt(nrl2,nnva)
          wzivt(irl2,1)=wznvt(nrl2,nnva)
          wzivt(irl1,1)=wznvt(nrl1,nnva)
          wrivt(irl21,1)=wrnvt(nrl21,nnva)
          wtivt(irl21,1)=wtnvt(nrl21,nnva)
          wzivt(irl21,1)=wznvt(nrl21,nnva)
c         wrivt(irl22,1)=wrnvt(nrl22,nnva)
c         wtivt(irl22,1)=wtnvt(nrl22,nnva)
c         wzivt(irl22,1)=wznvt(nrl22,nnva)
c    *
c    ***********************************************
c    *
c    *  INTERPOLATION OF THE 'OLD' AND THE 'NEW' DISTRIBUTIONS
c    *  OF VELOCITIES
c    *
          if(itrcw.ne.0) write(iwr,462)
462       format("1*** WNOFC *** : VELOCITIES AT THE NODES ")
c    *
c    *  WEIGHTING FACTORS FOR THE 'NEW' AND THE 'OLD' VELOCITIES
          frn=.4
          ftn=.7
          fzn=.3
          if(niter.ge.3) goto 49
          frn=.5
          ftn=.8
          fzn=.5
          if(niter.eq.2) goto 49
          frn=.5
          ftn=.9
```

```
      fzn=.5
49    continue
      if(itrcw.ne.0) write(iwr,951)frn,ftn,fzn
951   format(" *** WEIGHTING FACTORS: FRN,FTN,FZN ",3f6.2)
      frv=1.-frn
      ftv=1.-ftn
      fzv=1.-fzn
c  *
      do 47 j=1,nnva
      i1=nrl1
      i2=nrl2
 7003 do 43 i=i1,i2
      if(i.eq.nrl2) i3=0
      if(i.ne.nrl2) i3=1
      wrnv(i,j)=wrnvt(i,j)*frn+wrnv(i,j)*frv
      wtnv(i,j)=wtnvt(i,j)*ftn+wtnv(i,j)*ftv
      wznv(i,j)=wznvt(i,j)*fzn+wznv(i,j)*fzv
      go to 43
648   wrnv(i,j)=wrnvt(i,j)
      wtnv(i,j)=wtnvt(i,j)
      wznv(i,j)=wznvt(i,j)
43    continue
      if(itrcw.eq.0) goto 7002
      write(iwr,103)
      write(iwr,101)lrnv,(wrnv(i,j),i=1,nnvr)
      write(iwr,101)ltnv,(wtnv(i,j),i=1,nnvr)
      write(iwr,101)lznv,(wznv(i,j),i=1,nnvr)
 7002 i1=1
      i2=nnvr1
      if(i3.eq.0) go to 7003
47    continue
      do 48 j=1,niva
      do 44 i=irl1,irl2
      wriv(i,j)=wrivt(i,j)*frn+wriv(i,j)*frv
      wtiv(i,j)=wtivt(i,j)*ftn+wtiv(i,j)*ftv
      wziv(i,j)=wzivt(i,j)*fzn+wziv(i,j)*fzv
      go to 44
649   wriv(i,j)=wrivt(i,j)
      wtiv(i,j)=wtivt(i,j)
      wziv(i,j)=wzivt(i,j)
44    continue
      if(itrcw.eq.0) goto 48
      write(iwr,103)
      write(iwr,101)lriv,(wriv(i,j),i=irl1,irl2)
      write(iwr,101)ltiv,(wtiv(i,j),i=irl1,irl2)
      write(iwr,101)lziv,(wziv(i,j),i=irl1,irl2)
48    continue
c  *
c  *****************************************************************
c  *
c  *  INTERPOLATION OF THE INFLUENCE COEFFICIENTS
c  *
      if(niter.eq.1) return
      fzn=.5
      fzv=1.-fzn
      if(itrcw.eq.0) goto 90
      write(iwr,500)
500   format(" *** WNOFC: INTERPOLATION OF INF. COEFF ",/,
     &" INF. COEFF FROM LOOP2&")
      do 91 j=1,nncr2
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      write(iwr,101)ltwy,(twy(i,j),i=1,nncr2)
      write(iwr,101)ltwz,(twz(i,j),i=1,nncr2)
      write(iwr,103)
91    continue
      write(iwr,501)
501   format(" OLD INF. COEFF. ")
      do 92 j=1,nncr2
      write(iwr,101)ltwy,(twyt(i,j),i=1,nncr2)
      write(iwr,101)ltwz,(twzt(i,j),i=1,nncr2)
      write(iwr,103)
92    continue
90    continue
c   *
      do 82 i=1,nncr2
      do 82 j=1,nncr2
      twy(i,j)=fzn*twy(i,j)+fzv*twyt(i,j)
      twz(i,j)=fzn*twz(i,j)+fzv*twzt(i,j)
82    continue
c   *
      if(itrcw.eq.0) return
      write(iwr,502)fzn,fzv
502   format(" INT. INF. COEFF.: FZN=",e20.5," FZV=",e20.5)
      do 93 j=1,nncr2
      write(iwr,101)ltwy,(twy(i,j),i=1,nncr2)
      write(iwr,101)ltwz,(twz(i,j),i=1,nncr2)
      write(iwr,103)
93    continue
      return
c   *
101   format(1x,a4,1x,5f13.9,/,6x,5f13.9,/,6x,5f13.9)
102   format(1x,a4,1x,9i13,/,6x,9i13)
103   format(1x)
c   *
      end
```

```
      subroutine intgr2
c    *
c    *  **********************
c    *  *                    *
c    *  *    PROGRAM FWC     *
c    *  *                    *
c    *  *  SUBROUTINE INTGR2 *
c    *  *                    *
c    *  **********************
c    *
c    *     03/29/78
c    *
c    *  INTEGRATION OF THE VELOCITIES AT THE NODES ,
c    *  COMPUTED BY WNOFC2, TO GET THE NEW POSITIONS OF THE NODES
c    *
      common/all/cg1,cg2,cg3,cg4,ir
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nr11,nr12,ir11,ir12,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      common /wdata/xt(28),yt(28),zt(28),rin(28),vrin(28),vtin(28),
     &vz1(28),vx2(28),vy2(28),vz2(28)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nicr1,nicr2)
c    *
      etam=.5*(etanv(nnvr)+etanv(1))
c    *
      xep=0.
      do 13 kpt=1,nncr2
13    xep=xep+wzc(kpt)
      xep=xep/float(nncr2)
c    ***************************
c    *
c    *  VERIFICATION OF THE INTEGRATION
c    *
      isav=itrtg
      if(ivergr.ne.1) go to 91
      itrtg=1
      goto 91
92    continue
      ivergr=0
91    continue
c    *
c    ***************************
c    *
c    *  LOOP ON THE TWO WAKES
c    *  KW=1: INTEGRATION FOR THE NEAR WAKE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
c    *  KW=2: INTEGRATION FOR THE INTERMEDIATE WAKE
c    *
        do 90 kw=1,2
        if(kw.eq.2) goto 60
c    *
c    *  INITIALISATION FOR THE NEAR WAKE
c    *
        nr121=nr12-1
c       nr122=nr12-2
        ir121=ir12-1
c       ir122=ir12-2
        do 42 i=nr11,nr12
        if(i.eq.nr12) rin(i)=cg4
        if(i.eq.nr121) rin(i)=cg3
        if(i.eq.nr11) rin(i)=cg1
c       if(i.eq.nr122) rin(i)=cg2
        vrin(i)=wrnv(i,1)
        vtin(i)=wtnv(i,1)+rin(i)
         vzi(i)=wznv(i,1)+fmu
        xnv(i,1)=rin(i)
        ynv(i,1)=0.
        znv(i,1)=0.
        xt(i)=rin(i)
        yt(i)=0.
        zt(i)=0.
42      continue
        dpsi=dpsin*.5
        dzr=3.*abs((xep+fmu)*dpsin)
        nva=nnva
        nvr=nr12
        goto 61
c    *
c    *  INITIALISATION FOR THE INTERMEDIATE WAKE
60      continue
        do 52 i=ir11,ir12
        if(i.eq.ir11) i1=nr11
        if(i.eq.ir121) i1=nr121
c       if(i.eq.ir122) i1=nr122
        if(i.eq.ir12) i1=nr12
        xiv(i,1)=xnv(i1,ntva)
        yiv(i,1)=ynv(i1,ntva)
        ziv(i,1)=znv(i1,ntva)

        xt(i)=xiv(i,1)
        yt(i)=yiv(i,1)
        zt(i)=ziv(i,1)
        rin(i)=sqrt(xiv(i,1)**2+yiv(i,1)**2)
        vrin(i)=wriv(i,1)
        vtin(i)=wtiv(i,1)+rin(i)
        vzi(i)=wziv(i,1)+fmu
52      continue
        dpsi=dpsii*.5
c    *
        dzr=3.*abs((xep+fmu)*dpsii)
        nva=niva
        nvr=ir12
c    *
c    *
61      continue
100     format("1*** INTGR ***: KW=",i1," NVA=",i2," NVR=",i2," IK=",i2,
```

```
if
          &" DPSI=",f8.2)
c   *
c   * THE RADIAL AND TANGENTIAL VELOCITIES ARE WEIGHTED BY
c   * RNEW/ROLD.
c   * ALL THE POINTS ARE CONSTRAINED TO BE IN THE PLANE (P)
c   * WHICH CONTAINS THE NEW POINT OF THE REFERENCE STREAMLINE(IK)
c   * AND THE PROJECTION OF THIS POINT ON THE Z AXIS
c   * IN ORDER TO LIMIT THE DISTORSIONS OF THE ELEMENTS
c   *
c   * LOOP ON THE AZIMUTAL POSITIONS
c   *
        do 81 j=2,nva
c   *
c   * UNCORRECTED NEW POINTS (FIRST LOOP ON THE RADIAL POSITIONS)
        if(kw.eq.1) ij=nrl1
        if(kw.eq.2) ij=irl1
c   *
        do 82 i=ij,nvr
        if(kw.eq.2) goto 711
c   * OLD RADIUS OF THE POINT
        r2o=sqrt(xnv(i,j)**2+ynv(i,j)**2)
c   * RADIAL VELOCITY AT THE POINT (UNCORRECTED)
        vr2o=wrnv(i,j)
c   * TANGENTIAL VELOCITY AT THE POINT (UNCORRECTED)
        vt2o=wtnv(i,j)+r2o
c   * AXIAL VELOCITY
        vz2(i)=wznv(i,j)+fmu
        go to 712
        711 r2o=sqrt(xiv(i,j)**2+yiv(i,j)**2)
        vr2o=wriv(i,j)
        vt2o=wtiv(i,j)+r2o
        vz2(i)=wziv(i,j)+fmu
712     continue
c   *
c   * NEW RADIUS OF THE POINT:
c   * MEAN RADIAL VELOCITY:= .5*(VR1N+VR2N)
c   * VR1N: VELOCITY AT THE PRECEDENT POINT (CORRECTED)
c   * VR2N: VELOCITY AT THE CURRENT POINT (COREECTED)
c   * VR2N:=VR2O*R2N/R2O
        r2n=(r1n(i)+dpsi*vr1n(i))/(1.-dpsi*vr2o/r2o)
c   * CORRECTIONS ON THE RADIAL AND TANGENTIAL COMPONENTS
c   * OF THE VELOCITIES
        at=r2n/r2o
        if(at.gt.1.05)r2n=r2o*1.05
        if(at.gt.1.05)at=1.05
        vr2n=vr2o*at
        vt2n=vt2o*at
c   * NEW AZIMUTH (TEMPORARY)
        th2=atan2(yt(i),xt(i))+dpsi*2.*(vt1n(i)+vt2n)/(r1n(i)+r2n)
        cc=cos(th2)
        ss=sin(th2)
c   * RECTANGULAR COORDINATES OF THE UNCORRECTED NEW POINT
        xt(i)=r2n*cc
        yt(i)=r2n*ss
        ztd=dpsi*(vz2(i)+vz1(i))
        if(abs(ztd).gt.dzr) ztd=sign(dzr,ztd)
        zt(i)=zt(i)+ztd
c   * VELOCITY AT THIS POINT
        vx2(i)=vr2n*cc-vt2n*ss
        vy2(i)=vr2n*ss+vt2n*cc
```

```
c    *  SAVE RADIAL AND TANGENTIAL VELOCITIES, AND RADIAL COORDINATE
c    *  WILL BE USED FOR THE NEXT AZIMUTHAL POSITION
          vt1n(i)=vt2n
          vr1n(i)=vr2n
          vz1(i)=vz2(i)
          r1n(i)=r2n
82        continue
c    *
          do 83 i=ij,nvr
          if(kw.eq.2) goto 721
          xnv(i,j)=xt(i)
          ynv(i,j)=yt(i)
          znv(i,j)=zt(i)
          goto 83
721       continue
          xiv(i,j)=xt(i)
          yiv(i,j)=yt(i)
          ziv(i,j)=zt(i)
83        continue
81        continue
90        continue
c    *
          if(ivergr.eq.1) goto 92
          itrtg=isav
c    *
          return
c    *
          end
```

```
      subroutine intgr3
c   *
c   *  *********************
c   *  *                   *
c   *  *    PROGRAM FWC    *
c   *  *                   *
c   *  *  SUBROUTINE INTGR3 *
c   *  *                   *
c   *  *********************
c   *
c   *     03/29/78
c   *
c   *  INTEGRATION OF THE VELOCITIES AT THE NODES ,
c   *  COMPUTED BY WNOFC3, TO GET THE NEW POSITIONS OF THE NODES
c   *
      common/all/cg1,cg2,cg3,cg4,ir
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nr11,nr12,ir11,ir12,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      common /wdata/xt(28),yt(28),zt(28),rin(28),vrin(28),vtin(28),
     &vz1(28),vx2(28),vy2(28),vz2(28)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nicr1,nicr2)
c   *
      etam=.5*(etanv(nnvr1)+etanv(1))
c   *
      xep=0.
      do 13 kpt=1,nncr2
13    xep=xep+wzc(kpt)
      xep=xep/float(nncr2)
c   *************************
c   *
c   *  VERIFICATION OF THE INTEGRATION
c   *
      isav=itrtg
      if(ivergr.ne.1) go to 91
      itrtg=1
      goto 91
92    continue
      ivergr=0
91    continue
c   *
c   *************************
c   *
c   *  LOOP ON THE TWO WAKES
c   *  KW=1: INTEGRATION FOR THE NEAR WAKE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
c     *  KW=2: INTEGRATION FOR THE INTERMEDIATE WAKE
c     *
           do 90 kw=1,2
           if(kw.eq.2) goto 60
c     *
c     *  INITIALISATION FOR THE NEAR WAKE
c     *
           irl21=irl2-1
           nrl21=nrl2-1
           do 42 i=nrl1,nrl2
           if(i.eq.nrl2) rln(i)=cg4
           if(i.eq.nrl1) rln(i)=cg1
           if(i.eq.nrl21) rln(i)=cg3
           vrln(i)=wrnv(i,1)
           vtln(i)=wtnv(i,1)+rln(i)
            vzl(i)=wznv(i,1)+fmu
           xnv(i,1)=rln(i)
           ynv(i,1)=0.
           znv(i,1)=0.
           xt(i)=rln(i)
           yt(i)=0.
           zt(i)=0.
42         continue
           do 43 i=1,nnvrl
           if(etanv(i).gt.etam) go to 41
43         ik=i
41         if(etanv(ik+1)-etam.lt.etam-etanv(ik)) ik=ik+1
           do 50 i=1,nnvrl
           rln(i)=etanv(i)
           vrln(i)=wrnv(i,1)
           vtln(i)=wtnv(i,1)+etanv(i)
           vzl(i)=wznv(i,1)+fmu
           xnv(i,1)=etanv(i)
           ynv(i,1)=0.
           znv(i,1)=0.
           xt(i)=etanv(i)
           yt(i)=0.
           zt(i)=0.
50         continue
           dpsi=dpsin*.5
           dzr=3.*abs((xep+fmu)*dpsin)
           nva=nnva
           nvr=nrl2
           goto 61
c     *
c     *  INITIALISATION FOR THE INTERMEDIATE WAKE
60         continue
           do 52 i=irl1,irl2
           if(i.eq.irl1) il=nrl1
           if(i.eq.irl21) il=nrl21
           if(i.eq.irl2) il=nrl2
           xiv(i,1)=xnv(il,ntva)
           yiv(i,1)=ynv(il,ntva)
           ziv(i,1)=znv(il,ntva)

           xt(i)=xiv(i,1)
           yt(i)=yiv(i,1)
           zt(i)=ziv(i,1)
           rln(i)=sqrt(xiv(i,1)**2+yiv(i,1)**2)
           vrln(i)=wriv(i,1)
```

```
      vtin(i)=wtiv(i,1)+rin(i)
      vz1(i)=wziv(i,1)+fmu
52    continue
      dpsj=dpsii*.5
c  *
      dzr=3.*abs((xgg+fmu)*dpsii)
      nva=niva
      nvr=irl2
c  *
c  *
61    continue
100   format("1*** INTGR ***: KW=",i1," NVA=",i2," NVR=",i2," IK=",i2,
     6" DPSI=",f8.2)
c  *
c  *  THE RADIAL AND TANGENTIAL VELOCITIES ARE WEIGHTED BY
c  *  RNEW/ROLD.
c  *  ALL THE POINTS ARE CONSTRAINED TO BE IN THE PLANE (P)
c  *  WHICH CONTAINS THE NEW POINT OF THE REFERENCE STREAMLINE(IK)
c  *  AND THE PROJECTION OF THIS POINT ON THE Z AXIS
c  *  IN ORDER TO LIMIT THE DISTORSIONS OF THE ELEMENTS
c  *
c  *  LOOP ON THE AZIMUTAL POSITIONS
c  *
      do 81 j=2,nva
      i2=nvr
      i3=0.
c  *
c  *  UNCORRECTED NEW POINTS (FIRST LOOP ON THE RADIAL POSITIONS)
      if(kw.eq.1) i1=nrl1
      if(kw.eq.2) i1=irl1
c  *
1000  do 82 i=i1,i2
      if(i2.eq.nnvr1) i3=1
      if(kw.eq.2) goto 711
c  *  OLD RADIUS OF THE POINT
      r2o=sqrt(xnv(i,j)**2+ynv(i,j)**2)
c  *  RADIAL VELOCITY AT THE POINT (UNCORRECTED)
      vr2o=wrnv(i,j)
c  *  TANGENTIAL VELOCITY AT THE POINT (UNCORRECTED)
      vt2o=wtnv(i,j)+r2o
c  *  AXIAL VELOCITY
      vz2(i)=wznv(i,j)+fmu
      go to 712
      711 r2o=sqrt(xiv(i,j)**2+yiv(i,j)**2)
      vr2o=wriv(i,j)
      vt2o=wtiv(i,j)+r2o
      vz2(i)=wziv(i,j)+fmu
712   continue
c  *
c  *  NEW RADIUS OF THE POINT:
c  *  MEAN RADIAL VELOCITY:= .5*(VR1N+VR2N)
c  *  VR1N: VELOCITY AT THE PRECEDENT POINT (CORRECTED)
c  *  VR2N: VELOCITY AT THE CURRENT POINT (COREECTED)
c  *  VR2N:=VR2O*R2N/R2O
      r2n=(rin(i)+dpsi*vrin(i))/(1.-dpsi*vr2o/r2o)
c  *  CORRECTIONS ON THE RADIAL AND TANGENTIAL COMPONENTS
c  *  OF THE VELOCITIES
      at=r2n/r2o
      if(at.gt.1.05)r2n=r2o*1.05
      if(at.gt.1.05)at=1.05
```

```
            vr2n=vr2o*at
            vt2n=vt2o*at
c    *  NEW AZIMUTH (TEMPORARY)
            th2=atan2(yt(i),xt(i))+dpsi*2.*(vt1n(i)+vt2n)/(r1n(i)+r2n)
            cc=cos(th2)
            ss=sin(th2)
c    *  RECTANGULAR COORDINATES OF THE UNCORRECTED NEW POINT
            xt(i)=r2n*cc
            yt(i)=r2n*ss
            ztd=dpsi*(vz2(i)+vz1(i))
            if(abs(ztd).gt.dzr) ztd=sign(dzr,ztd)
            zt(i)=zt(i)+ztd
c    *  VELOCITY AT THIS POINT
            vx2(i)=vr2n*cc-vt2n*ss
            vy2(i)=vr2n*ss+vt2n*cc
c    *  SAVE RADIAL AND TANGENTIAL VELOCITIES, AND RADIAL COORDINATE
c    *  WILL BE USED FOR THE NEXT AZIMUTHAL POSITION
            vt1n(i)=vt2n
            vr1n(i)=vr2n
            vz1(i)=vz2(i)
            r1n(i)=r2n
82          continue
            if(i2.eq.nnvr1) go to 1001
            go to 1003
1001        x0=xt(ik)
            y0=yt(ik)
            z0=zt(ik)
            vx=vx2(ik)
            vy=vy2(ik)
            vz=vz2(ik)
            alpha=-(x0*vx+y0*vy)/(x0*x0+y0*y0)
            vx=vx+alpna*x0
            vy=vy+alpha*y0
            do 85 i=i1,i2
            beta=((x0-xt(i))*vx+(y0-yt(i))*vy+(z0-zt(i))*vz)/
           &(vx2(i)*vx+vy2(i)*vy+vz2(i)*vz)
            xt(i)=xt(i)+beta*vx2(i)
            yt(i)=yt(i)+beta*vy2(i)
            zt(i)=zt(i)+beta*vz2(i)
            r1n(i)=sqrt(xt(i)**2+yt(i)**2)
            vz1(i)=vz2(i)
            xnv(i,j)=xt(i)
            ynv(i,j)=yt(i)
            znv(i,j)=zt(i)
85          continue
            go to 81
1003        do 83 i=i1,i2
            if(kw.eq.2) goto 721
            xnv(i,j)=xt(i)
            ynv(i,j)=yt(i)
            znv(i,j)=zt(i)
            goto 83
721         continue
            xiv(i,j)=xt(i)
            yiv(i,j)=yt(i)
            ziv(i,j)=zt(i)
83          continue
            i1=1
            i2=nnvr1
            if(i3.eq.0.and.kw.eq.1) go to 1000
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
81        continue
90        continue
c    *
          if(ivergr.eq.1) goto 92
          itrtg=isav
c    *
          return
c    *
          end
```

```
      subroutine loop12
c   *
c   *  ********************
c   *  *                  *
c   *  *   PROGRAM FWC    *
c   *  *                  *
c   *  *  SUBROUTINE LOOP12 *
c   *  *                  *
c   *  ********************
c   *
c   *  07/14/80
c   *
c   *  EVALUATION OF THE INDUCED VELOCITIES AND OF THE DISTRIBUTION
c   *  OF CIRCULATION ALONG THE BLADES,
c   *  FROM THE INFLUENCE COEFFICIENTS RETURNED BY LOOP2
c   *
c   *  THE SUBROUTINE RETURNS:
c   *  ITEST=0 IF CONVERGENCE IS REACHED
c   *  ITEST=1 IF CONVERGENCE IS REACHED FOR THE LOOP ON THE CIRCULATION
c   *  ITEST=2 IF CONVERGENCE IS NOT REACHED WITHIN LIM1 ITERATIONS
c   *  ON THE CIRCULATION
c   *
c   *  TWX(I,J) CONTAINS THE DERIVATIVE :
c   *  D(WX(I))/D(GAMMA(J+1))
c   *  TWY: SAME FOR THE Y COMPONENT
c   *  TWZ: SAME FOR THE Z COMPONENT
c   *  NOTE :  THE SUBSCRIPTS I AND J ARE REFERED TO THE INNER
c   *  CENTERS OF THE ELEMENTS
c   *
      common/roll5/wxctr(25),wyctr(25),wzctr(25)
      common/roll10/wxctr2(25),wyctr2(25),wzctr2(25)
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,isamet,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nblds1,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,
     &nnvr,nnva,nnvr1,nnva1,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivr1,niva1,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntva1,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imodel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      dimension wyct(24),wzct(24)
      dimension wyctt(24),wzctt(24)
      dimension gtemp(26)
      dimension mtest(24)
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      data pi2/9.869604404/
  data pi/3.141592653/
      data lx,ly,lz/"WXC ","WYC ","WZC "/
      data lg,lyt,lzt/"GMC ","WYCT","WZCT"/
      data lts/"TEST"/
c   *
```

```
          ltest=0
c    *
          do 80 i=1,nncr
          gtemp(i+1)=gamme(i+1)
          wyctt(i)=wyc(i)
          wzctt(i)=wzc(i)
80        continue
          gtemp(nncr1)=0.0
c    *
          do 90 iter=1,lim1
c    *
c    *  EVALUATE THE DISTRIBUTION OF CIRCULATION AND EPS (TEST VALUE)
c    *
          eps=0.
          nncr3=nncr2-1
c
c           TIP LOSS FACTOR
c
          gamme(nncr1)=0.0
          do 1 i=1,nncr2
          eps=eps+wzc(i)
          tlam=(wzc(i)+fmu)/(etanc(i+1)+wyc(i))
          flamda=atan(tlam)
          if(flamda.lt.-.15) flamda=-.15
          u=(etanc(i+1)+wyc(i))/cos(flamda)
          alp=flamda-thetac(i)
          if(abs(alp).gt.alphas.and.alp.gt.0.) alp=alphas
          if(abs(alp).gt.alphas.and.alp.lt.0.) alp=-alphas
          if(i.eq.nncr2) go to 1
          gamme(i+1)=pi*pi*1.*sigma*u*alp/blades
1         continue
c    *
c         TIP LOSS FACTOR
c
c    * TEST VALUE: .01 TIMES THE MEAN Z INDUCED VELOCITY
          eps=.02*abs(eps)/nncr2
          epp=1.5*eps
          epp2=3.*eps
c    *
          ktest=1
          nncr3=nncr2-1
          do 2 i=1,nncr2
          wyct(i)=0.
          wzct(i)=0.
          wzctr(i)=0.
          wyctr(i)=0.
          do 3 j=1,nncr2
          wyct(i)=wyct(i)+twy(i,j)*gamme(j+1)
          wzct(i)=wzct(i)+twz(i,j)*gamme(j+1)
3         continue
2         continue
          call loop0
          do 1000 i=1,nncr2
          wyct(i)=wyct(i)+wyctr(i)
1000      wzct(i)=wzct(i)+wzctr(i)
          do 1100 i=1,nncr2
          if(abs(wyct(i)-wyc(i)).gt.epp) ktest=0
          if(abs(wzct(i)-wzc(i)).gt.epp) ktest=0
1100      continue
c    *
```

```
      if(itrcl1.eq.0) goto 60
      write(iwr,100)iter
100   format(/" *** LOOP1 ***: NITER=",i2)
      write(iwr,101)ly ,(wyc (kk),kk=1,nncr2)
      write(iwr,101)lz ,(wzc (kk),kk=1,nncr2)
      write(iwr,101)lg ,(gammc(kk),kk=2,nncr1)
      write(iwr,101)lyt,(wyct(kk),kk=1,nncr2)
      write(iwr,101)lzt,(wzct(kk),kk=1,nncr2)
      write(iwr,150) eps,epp
 150  format("eps=",f10.5,10x,"epp= ",f10.5)
101   format(1x,a4,1x,6f10.6,/,6x,6f10.6)
c   *
60    continue
c   *
c   *  CONVERGENCE IS REACHED: GOTO 50
      if(ktest.eq.1) goto 50
c   *
c   *  NEW APPROXIMATION OF THE INDUCED VELOCITIES
c   *
      facn=.2
      facv=1.-facn
c   *
      do 6 i=1,nncr2
      wyc(i)=wyc(i)*facv+wyct(i)*facn
      wzc(i)=wzc(i)*facv+wzct(i)*facn
6     continue
90    continue
c   *
c   *  NO CONVERGENCE WITHIN LIM1 ITERATIONS
c   *
      itest=2
      if(lim1.eq.1) itest=1
      goto 25
c   *
c   ************************
c   *
50    continue
      if(lim1.eq.1) go to 25
c   *
c   *  TEST OF CONVERGENCE FOR THE OUTER LOOP:
c   *  THE CONVERGED VALUE OF THE INDUCED VELOCITIES
c   *  MUST BE WITHIN (EPS) OF THE ORIGINAL INDUCED VELOCITIES
c   *
      ktest=1
      if(itrcl1.eq.1) write(iwr,150)eps
      do 51 i=1,nncr2
      mtest(i)=0
      if(abs(wyc(i)-wyctt(i)).lt.epp2) go to 52
      mtest(i)=1
      ktest=0
52    continue
      if(abs(wzc(i)-wzctt(i)).lt.epp2) go to 51
      mtest(i)=mtest(i)+2
      ktest=0
51    continue
      if(itrcl1.ne.0) write(iwr,110)lts,(mtest(kk),kk=1,nncr2)
110   format(1x,a4,1x,9i10,/,6x,9i10)
      if(ktest.eq.1) goto 25
      itest=1
c   *
```

```
c    *  NEW APPROXIMATION OF THE INDUCED VELOCITIES
c    *  AND OF THE DISTRIBUTION OF CIRCULATION
c    *
           facn=.5
           if(niter.le.2) facn=.1
           facv=1.-facn
           if(itrcl1.ne.0) write(iwr,200) facn
200        format(" *** LOOP1 *** OLD/NEW FACT: FACN=",f10.5)
           do 91 i=2,nncr1
           gammc(i)=facn*gammc(i)+facv*gtemp(i)
91         continue
c          TIP LOSS FACTOR
c
           gammc(nncr1)=0.0
c
           if(itrcl1.eq.1) write(iwr,101) lg, (gammc(kk),kk=2,nncr1)
           do 7 i=1,nncr2
           wyc(i)=0.
           wzc(i)=0.
           wxctr(i)=0.
           wyctr(i)=0.
           wzctr(i)=0.
           do 8 j=1,nncr2
           wyc(i)=wyc(i)+twy(i,j)*gammc(j+1)
           wzc(i)=wzc(i)+twz(i,j)*gammc(j+1)
8          continue
7          continue
           call loop0
           do 11 i=1,nncr2
           wyc(i)=wyc(i)+wyctr(i)
           wzc(i)=wzc(i)+wzctr(i)
11         continue
c    *
25         continue
c    *
c    *  X COMPONENTS OF THE INDUCED VELOCITIES
c    *
           do 14 i=1,nncr2
           wxc(i)=0.
           wxctr(i)=0.
           do 10 j=1,nncr2
           wxc(i)=wxc(i)+twx(i,j)*gammc(j+1)
10         continue
14         continue
           call loop0
           do 12 i=1,nncr2
           wxc(i)=wxc(i)+wxctr(i)
12         continue
           if(itrcl1.eq.0) goto 9
           write(iwr,111) iter
111        format(/,"*** LOOP1 *** LAST ITERATION:",i3)
           write(iwr,101) lx , (wxc  (kk),kk=1,nncr2)
           write(iwr,101) ly , (wyc  (kk),kk=1,nncr2)
           write(iwr,101) lz , (wzc  (kk),kk=1,nncr2)
           write(iwr,120)
           write(iwr,101) lyt, (wyctt(kk),kk=1,nncr2)
           write(iwr,101) lzt, (wzctt(kk),kk=1,nncr2)
           write(iwr,120)
           write(iwr,101) lg, (gammc(kk),kk=2,nncr2)
           write(iwr,121) epp
```

```
121       format(/," EPP=",f10.6,/)
9         continue
c   *
          if(itrace.eq.1.or.itrcl1.eq.1) write(iwr,103)itest
103       format(" *** LOOP1 *** : RETURN CODE : ",i3)
          if(itest.eq.2) call out2(1)
          if(itest.eq.2) write(iwr,104)liml
104       format(" *** LOOP1 *** : NO CONVERGENCE WITHIN ",i3," ITERATIONS")
120       format(1x)
          write(6,7000) (gammc(i),i=2,nncri)
7000      format(5x,"GAMMC=",20f10.5)
          write(6,5000) znv(13,13),xnv(13,13),ynv(13,13)
 5000 format(/,5x,'ZNV= ',f10.5,'XNV= ',f10.5,'YNV= ',f10.5)
          return
          end
```

```
c*
      subroutine loop0
c*
c*
      common/roll5/wxctr(25),wyctr(25),wzctr(25)
      common/roll10/wxctr2(25),wyctr2(25),wzctr2(25)
      common/semi/delz
      common/parm/iwr,ird,itrace,ipsem,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isameb,iscmat,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/geom/nbldsl,nblds,sigma,fmu,etan(25),knnvr,etai(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeff1,c,s,blades,nnvr,
     &nnva,nnvrl,nnval,nncr,nnca,etanv(25),etanc(26),nivr,niva,nivrl,nival,
     &nicr,nica,etaiv(6),etaic(7),ntva,ntval,ntca,fpsi,fps2,nrl1,nrl2,irl1,
     &irl2,imodel
      common/resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
      &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),ia,ib
      common/vel/x,y,z,ux,uy,uz,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,
     &gm,gm1,gm2,dgm,it,rho,zpf,t1,eps1,eps2,stra,jgm
      data twopi/6.283185306/
c*
c*    THIS SUBROUTINE COMPUTES THE INDUCED VELOCITIES ON THE BLADE
c*    DUE TO INTERMEDIATE AND FAR WAKE ROLLED UP VORTICES
c*
      nncr2=nncr-2
       nncr5=nncr2+3
      do 1000 j=3,nncr2
      k=nncr5-j
      if(abs(gammc(k)).lt.abs(gammc(k-1))) go to 1000
      go to 2000
1000  continue
2000  q4=-gammc(k)
      q3=gammc(k)-gammc(k-3)
      q1=gammc(k-3)-gammc(2)
      it=2
      irl21=irl2-1
c     irl22=irl2-2
      nrl21=nrl2-1
c     nrl22=nrl2-2
      do 3000 i=irl1,irl2
      if(i.eq.irl1) gm=q1
c     if(i.eq.irl22) gm=q2
      if(i.eq.irl21) gm=q3
      if(i.eq.irl2) gm=q4
      do 3100 j=1,nival
      dps2=eps2
      x1=xiv(i,j)
      y1=yiv(i,j)
      z1=ziv(i,j)
      x3=xiv(i,j+1)
      y3=yiv(i,j+1)
      z3=ziv(i,j+1)
      phib=twopi/blades
      kapa=ifix(phib/dpsii)
```

```
      nvsp=niva-kapa
      phil=dpsin*(nnva-1)+dpsii*(j-1)
      if(phil.gt.phib) eps2=.1
      call coord
      if(itrans.eq.0.or.j.gt.2) call vindb2(4,1)
      eps2=dps2
3100  continue
3000  continue
c*
c*    FOR 2 SPIRALS ; 720 degrees
c*
      do 3200 i=ir11,ir12
      if(i.eq.ir11) gm=q1*1.
c     if(i.eq.ir122) gm=q2*1.
      if(i.eq.ir121) gm=q3*1.
      if(i.eq.ir12) gm=q4*1.
      zpf=ziv(i,niva)*2-ziv(i,nvsp)
      delz=ziv(i,niva)-ziv(i,nvsp)
      rho=.5*sqrt((xiv(i,niva)+xiv(i,niva1))**2+(yiv(i,niva)+
     &yiv(i,niva1))**2)
      it=3
      call vindb2(4,1)
3200  continue
      return
      end
```

```
      subroutine out2(iout)
c  *
c  *  *******************
c  *  *                 *
c  *  *   PROGRAM FWC   *
c  *  *                 *
c  *  *  SUBROUTINE OUT *
c  *  *                 *
c  *  *******************
c  *
c  *     02/14/78
c  *
c  *  OUTPUT OF THE RESULTS
c  *
c  *  ENTRY OUTINT: OUTPUT FOR INTERMEDIATE RESULTS
c  *
      reallip,lt,lp,lpi
      common/parm/iwr,ird,itrace,lpscm,nmes,itrcw,itrcl1,itrcl2,itrcg,
     &itrcnt,itrcf,iplot,icgen,iplotv,itrtg,ivergr,itest,isamcb,isamat,
     &lims,lim1,lim2,niter,itrans,ipn,jpunch,iplotw,isw,icont,iplotg,
     &ipr,iplott
      common/hparm/ifmu
      common/geom/nbldsl,nbld*,sigma,fmu,etan(25),knnvr,etal(6),knivr,
     &itwist,thetad(25),theta(25),thetac(24),theta0,thet0d,alphas,
     &cd0,cdk,dpsind,dpsin,dpsiid,dpsii,coeff,coeffl,c,s,blades,
     &nnvr,nnva,nnvrl,nnval,nncr,nnca,etanv(25),etanc(26),
     &nivr,niva,nivrl,nival,nicr,nica,etaiv(6),etaic(7),
     &ntva,ntval,ntca,fps1,fps2,nrl1,nrl2,irl1,irl2,imcdel
      common /resul/gammc(26),twx(24,24),twy(24,24),twz(24,24),
     &wxc(24),wyc(24),wzc(24),wxnc(26,19),wync(26,19),wznc(26,19),
     &wxic(9,51),wyic(9,51),wzic(9,51),wrnv(28,18),wtnv(28,18),
     &wznv(28,18),wriv(8,50),wtiv(8,50),wziv(8,50),
     &xnc(26,19),ync(26,19),znc(26,19),xic(9,51),yic(9,51),zic(9,51),
     &xnv(28,18),ynv(28,18),znv(28,18),xiv(8,50),yiv(8,50),ziv(8,50),
     &ia,ib
      common /otdata/psi(28),r(28),gmc(24),
     &wxt(25),wyt(25),flamda(24),u(24),alpha(24),lip(24),flip(24),
     &dp(24),fdp(24),tdp(24),tp(24),fp(24),tlip(24),xt(28),yt(28),
     &wt(28),wr(28)
      common/temp/lt1
      equivalence (nnvr,nncr1),(nivr,nicr1),(nnva,nnca1)
      equivalence (niva,nica1),(nnvr1,nncr2),(nivr1,nicr2)
      data pi2/9.8696044/
      data pi/3.141592653/
      data leta,lwxc,lwyc,lwzc/"ETA ","WXC ","WYC ","WZC "/
      data lu  ,lgmc,llam,lalp/"U   ","GMC ","LAM ","ALP "/
      data lthe,llip,ltlp,lflp/"THE ","LIP ","TLIP","FLIP"/
      data ldp ,ltdp,lfdp,ltp /"DP  ","TDP ","FDP ","TP  "/
      data letn,lxn ,lyn ,lzn /"ETAN","XNV ","YNV ","ZNV "/
      data leti,lxi ,lyi ,lzi /"ETAI","XIV ","YIV ","ZIV "/
      data lwrn,lwtn,lwzn/"WRNV","WTNV","WZNV"/
      data lwri,lwti,lwzi/"WRIV","WTIV","WZIV"/
      data lr  ,lpsi/"R   ","PSI "/
      data lfp/"FP  "/
      data ltwy,ltwz/"TWY ","TWZ "/
c  *
110   format(1x,a4,1x,9f10.6,/,6x,9f10.6)
111   format(1x,a4,1x,9f10.3,/,6x,9f10.3)
120   format(1x)
140   format(i4)
```

ORIGINAL PAGE IS OF POOR QUALITY

```
c    *
        write(iwr,100)
100     format(10x,//,"FREE WAKE ANALYSIS")
        write(iwr,196) cd0,cdk
196     format(//,10x,"(CD0=",f5.3,", CDK=",f5.3,")")
        goto 40
c    *
c    ***************************************************
c    *
        entry outint2
c    *
        iout=1
        write(iwr,101)niter
101     format(1h1,15x,"FREE WAKE ANALYSIS  ",
         &" (INTERMEDIATE RESULTS, ITERATION NUMBER:",i3,1h))
c    *
40      continue
c    *
c    *
        lt=0.
        lp=0.
        lpi=0.
c    *
c       TIP LOSS FACTOR
c
        gammc(nncr1)=0.0
c
        do 1 i=1,nncr2
        tlam=(wzc(i)+fmu)/(etanc(i+1)+wyc(i))
        flamda(i)=atan(tlam)
        u(i)=(etanc(i+1)+wyc(i))/cos(flamda(i))
        alpha(i)=flamda(i)-thetac(i)
        alphal=alpha(i)
        if(abs(alpha(i)).gt.alphas.and.alpha(i).gt.0.) alphal=alphas
        if(abs(alpha(i)).gt.alphas.and.alpha(i).lt.0.) alphal=-alphas
        if(i.eq.nncr2) go to 1000
        gmc(i)=pi*pi*sigma*1.*u(i)*alphal/blades
1000    cd=cd0+cdk*alpha(i)**2
        if(abs(alpha(i)).gt.alphas)cd=2.*cd0+cdk*alpha(i)**2
        lip(i)=u(i)*gmc(i)
        tlip(i)=lip(i)*cos(flamda(i))
        flip(i)=-lip(i)*sin(flamda(i))
        dp(i)=u(i)*u(i)*(pi*sigma/blades)*cd*.5
        fdp(i)=dp(i)*cos(flamda(i))
        tdp(i)=dp(i)*sin(flamda(i))
        tp(i)=tlip(i)+tdp(i)
        fp(i)=flip(i)+fdp(i)
        lt=lt+tp(i)*(etanv(i+1)-etanv(i))
        lp=lp+fp(i)*(etanv(i+1)**2-etanv(i)**2)
        lpi=lpi+flip(i)*(etanv(i+1)**2-etanv(i)**2)
1       continue
        lt=lt*blades/pi
        lp=lp*blades/(pi*2.)
        lpi=lpi*blades/(pi*2.)
        tap=.01*lt
        tapp=lt1-lt
        if(abs(tapp).lt.abs(tap)) write(iwr,105)
105     format(//," CONVERGENCE FOR LT REACHED",/)
        lt1=lt
        if(iimu.eq.0) go to 5
```

```
      cp=lp/fmu**3
      ct=lt/fmu**2
      ctcp=ct/cp
5     continue
      write(iwr,110)leta,(etanc(i),i=2,nncr1)
      write(iwr,120)
      write(iwr,110)lwxc,(wxc(i),i=1,nncr2)
      write(iwr,110)lwyc,(wyc(i),i=1,nncr2)
      write(iwr,110)lwzc,(wzc(i),i=1,nncr2)
      write(iwr,120)
      write(iwr,110)lu,(u(i),i=1,nncr2)
      write(iwr,110)lgmc,(gmc(i),i=1,nncr2)
      write(iwr,120)
      write(iwr,110)llam,(flamda(i),i=1,nncr2)
      write(iwr,110)lalp ,(alpha(i),i=1,nncr2)
      write(iwr,110)ltha,(thetac(i),i=1,nncr2)
      write(iwr,120)
      write(iwr,110)llip,(lip(i),i=1,nncr2)
      write(iwr,110)ltlp,(tlip(i),i=1,nncr2)
      write(iwr,110)lflp,(flip(i),i=1,nncr2)
      write(iwr,110)ldp ,(dp  (i),i=1,nncr2)
      write(iwr,110)ltdp ,(tdp(i),i=1,nncr2)
      write(iwr,110)lfdp ,(fdp(i),i=1,nncr2)
      write(iwr,110)ltp  ,(tp (i),i=1,nncr2)
      write(iwr,110)lfp  ,(fp (i),i=1,nncr2)
      write(iwr,120)
      if(ifmu.ne.0) write(iwr,88) lt,lp,ct,cp,ctcp
      if(ifmu.eq.0) write(iwr,88) lt,lp
88    format(///,10x,"LT =",e12.5/10x,"LP =",e12.5/10x,"CT =",e12.5
     &/10x,"CP&=",e12.5//9x"CT/CP=",e12.5//)
      write(iwr,89) lpi
 89   format(///,10x,"LPI = ",e12.5)
      write(iwr,109)
109   format(1h1)
c   *
      if(iout.eq.2) return
c   *
115   continue
      iw=1
      nvr=nr12
      write(iwr,110)letn,(etanv(i),i=1,nnvr)
      do 10 j=1,nnva
      write(iwr,140)j
      write(iwr,110)lxn ,(xnv(i,j),i=nr11,nr12)
      write(iwr,110)lyn ,(ynv(i,j),i=nr11,nr12)
      write(iwr,110)lzn ,(znv(i,j),i=nr11,nr12)
      write(iwr,120)
      write(iwr,110)lwrn,(wrnv(i,j),i=nr11,nr12)
      write(iwr,110)lwtn,(wtnv(i,j),i=nr11,nr12)
      write(iwr,110)lwzn,(wznv(i,j),i=nr11,nr12)
      do 12 i=nr11,nr12
      wr(i)=wrnv(i,j)
      wt(i)=wtnv(i,j)
      xt(i)=xnv(i,j)
      yt(i)=ynv(i,j)
12    continue
      goto 30
11    continue
10    continue
c   *
```

```
        iw=2
        nvr=irl2
        write(iwr,110)leti,(etaiv(i),i=1,nivr)
        do 20 j=1,niva
        write(iwr,140)j
        write(iwr,110)ixi  ,(xiv(i,j),i=irl1,irl2)
        write(iwr,110)lyi  ,(yiv(i,j),i=irl1,irl2)
        write(iwr,110)lzi  ,(ziv(i,j),i=irl1,irl2)
        write(iwr,120)
        write(iwr,110)lwri,(wriv(i,j),i=irl1,irl2)
        write(iwr,110)lwti,(wtiv(i,j),i=irl1,irl2)
        write(iwr,110)lwzi,(wziv(i,j),i=irl1,irl2)
        do 22 i=irl1,irl2
        wr(i)=wriv(i,j)
        wt(i)=wtiv(i,j)
        xt(i)=xiv(i,j)
        yt(i)=yiv(i,j)
22      continue
        goto 30
21      continue
20      continue
443     format(8f10.5)
444     format(9f8.4)
445     format(6f10.5)
c   *
        write(iwr,203)
        do 93 i=1,nncr2
        write(iwr,201)ltwy,(twy(i,k),k=1,nncr2)
        write(iwr,201)ltwz,(twz(i,k),k=1,nncr2)
        write(iwr,202)
93      continue
201     format(1x,a4,1x,9f13.9/6x,9f13.9)
202     format(//)
203     format(" INFLUENCE COEFF.  TW(I,K)*GAMMA(K)=WI(I)"/)
c*
        if(imodel.ne.2) go to 204
        do 55 j=1,nnva
        write(iwr,140)j
        write(iwr,110)lxn ,(xnv(i,j),i=1,nnvr)
        write(iwr,110)lyn ,(ynv(i,j),i=1,nnvr)
        write(iwr,110)lzn ,(znv(i,j),i=1,nnvr)
        write(iwr,120)
        write(iwr,110)lwrn,(wrnv(i,j),i=1,nnvr)
        write(iwr,110)iwtn,(wtnv(i,j),i=1,nnvr)
        write(iwr,110)lwzn,(wznv(i,j),i=1,nnvr)
        do 56 i=1,nnvr
        wr(i)=wrnv(i,j)
        wt(i)=wtnv(i,j)
        xt(i)=xnv(i,j)
        yt(i)=ynv(i,j)
        r(i)=sqrt(xt(i)**2+yt(i)**2)
        cc=xt(i)/r(i)
        ss=yt(i)/r(i)
        phi=atan(yt(i)/xt(i))
        if(xt(i).lt.0.) phi=phi+pi
        if((xt(i).ge.0.).and.(yt(i).lt.0.)) phi=phi+pi+pi
        psi(i)=57.296*phi
56      continue
        write(iwr,120)
        write(iwr,111)lpsi,(psi(i),i=1,nnvr)
```

```
      write(iwr,110)ir ,(r(i),i=1,nnvr)
      write(iwr,120)
 55   continue
204   return
c  *
 30   continue
      if(iw.eq.1) ij=nr11
      if(iw.eq.2) ij=ir11
      do 31 i=ij,nvr
      r(i)=sqrt(xt(i)**2+yt(i)**2)
      cc=xt(i)/r(i)
      ss=yt(i)/r(i)
      phi=atan(yt(i)/xt(i))
      if(xt(i).lt.0.) phi=phi+pi
      if((xt(i).ge.0.).and.(yt(i).lt.0.)) phi=phi+pi+pi
      psi(i)=57.296*phi
 31   continue
      write(iwr,120)
      write(iwr,111)ipsi,(psi(i),i=ij,nvr)
      write(iwr,110)ir ,(r(i),i=ij,nvr)
      write(iwr,120)
      if(iw.eq.1) goto 11
      goto 21
c*
c  *
      end
```

TABLE 2

Subscripted Variables

etanv	Radial positions of the blade stations and the near wake vortices before they deform
etanc	Radial positions of the blade centers
gammc/Gmc	The bound circulation distribution
gammt1, gammt2	Strength of the root and tip vortex in the near wake, due to a unit circulation distribution
gm1n, gm2n	First and second lumped strengths of the vortex sheet elements in the near wake
hammt1, hammt2	Coefficients for gammt1, gammt2 respectively Note: by definition: gammt1(i) = Σ hammt1 (i,j)* gammc (j+1)
hm1n, hm2n	Coefficients for gm1n, gm2n respectively. Note: by definition: gm1n(i) = Σ hm1n (i,j) * gammc(j+1)
twx, twy, twz	X,y,z components of the influent coefficient. They are the induced velocities at the blade centers due to a unit bound circulation

wriv wtiv wziv	Radial, tangential and axial components of the induced velocity at node (i,j) of the intermediate wake vortices
wrnv wtnv wznv	Same as wriv, wtiv, wziv, but for intermediate wake vortices
wxc wyc wzc	X,y,z components of the total induced velocity at the blade centers. For example: wxc(i) = wxctr(i) + Σ twx(i,j) * gammc(j+1)
wxctr wyctr wzctr	X,y,z components of the induced velocity at the blade centers due to the intermediate and far wake vortices
wxivr wyivr wzivr	X,y,z components of the induced velocity at rolled up vortices in the intermediate wake
wxnc wync wznc	X,y,z components of the induced velocity at centers of the near wake (at coordinates xnc,ync, znc)
wxnvr wynvr wznvr	X,y,z components of the induced velocity at the anticipated rolled up vortices in the near wake

xiv, yiv, ziv	X,y,z cartesian coordinates of the node of intermediate wake vortex
xnc, ync, znc	X,y,z cartesian coordinates of the center of near wake vortex
xnv, ynv, znv	Same as xiv, yiv, ziv but now these are for the near wake. Xnv is the x coordinate of the node, located on the stream lines originating from the blade stations (etanv)

variables	
alphas	stall angle (in radians)
capk	elliptical integral of the first kind
cdo, cdk	drag coefficients (cd = cdo + α^2 cdk)
cg1	spanswise location of the rolled up root vortex
cg2	spanswise location of the rolled up 2nd middle vortex at $\phi = 0$
cg3	spanswise location of the rolled up 1st middle vortex
cg4	spanswise location of the rolled up tip vortex
coeff1/coeff	parameter of root and tip vortices (parameter k)
delz	Vertical spacing between the last node of the intermediate wake and the starting location of the far wake.
dpsiid	azimuthal size of each vortex element in the intermediate wake
dpsind	same as dpsiid but for the near wake vortex element
e	elliptical integral of the second kind

eps1	thickness of the rectangular vortex element
eps2	core radius of the segment element
etan	same as etanv
fmu	advance ratio
fps1	same as eps1
fps2	same as eps2

imodel
- = 2 : second vortex sheet model
- = 3 : third vortex sheet model
- = 4 : fourth vortex sheet model

irl1	rolled up root vortex	
irl2	rolled up tip vortex	all in the
irl21	rolled up first middle vortex	intermediate
irl22	rolled up second middle vortex	wake

it
- = 1 : the vortex is a sheet/rectangular element
- = 2 : the vortex is a filament/segment element
- = 3 : the vortex is a semi infinite cylinder

itest
- = 0 : convergence is reached
- = 1 : convergence is not reached, but the iteration on the bound circulation has converged
- = 2 : the iteration on the bound circulation does not converge within the liml limit.

kunvr	same as nnvr.
lim 1	limit of the number of iterations on the bound circulation
lim 2	limit of the number of iterations on the wake geometry
Ltwist	=1; twisted blade =0; untwisted blade
nblds	= number of rotor blades
niva	= number of aximuthal nodes in the intermediate wake
niter	= current loop index for the iteration on the wake geometry
nnca	= it is the number of centers in the azimuthal direction of the near wake (equal to nnva +1)
nnca1	equals to nnca -1, equivalent to nnv 2
nncr	it is the number of centers in the radial direction of the blade and near wake.
nncr1	equals to nncr -1, equivalent to nnvr
nncr2	equals to nncr -2, equivalent to nnvr 1

nnva — number of nodes in the azimuthal direction of the near wake.

nnva1 — equals to nnva -1

nnvr — number of stations/nodes in the radial direction of the blade and the near wake.

nnvr1 — equals to nnvr -1. It is the number of vortex elements of the near wake (due to tip loss)

nr11	anticipated rolled up root vortex	in the near wake
nr12	anticipated rolled up tip vortex	
nr121	anticipated rolled up first middle vortex	
nr122	anticipated rolled up second middle vortex	

Note: nr11 must be higher than nnvr. If there are only 3 rolled up vortices, then the nr122 is not used.

ntva — same as nnva

q1 — strength of the rolled up root vortex

q2 — strength of the rolled up second middle vortex

q3 — strength of the rolled up first middle vortex

q4 — strength of the rolled up tip vortex

rho — radius of a vortex cylinder in the far wake

sigma	solidity of rotor
thetad	the pitch angle at each spanswise station (in degrees) For untwisted blade, thetad equals to the pitch angle (in degrees) must be specified.
ux uy uz	x,y,z components of the induced velocity due to one vortex element.
wi	vertical velocity induced at a segment element due to a vortex ring (excluding the self-contribution).
wil	lamb vortex ring self-induced velocity.
wis	equals to wil-wi, is the self-element vertical induced velocity.
x,y,z	point at which the induced velocities are computed
x_1,y_1,z_1	coordinates of the first point on the rectangular on the segment element
x_2,y_2,z_2	coordinates of the second point on the rectangular element
x_3,y_3,z_3	coordinates of the third point on the rectangular element or the second point on the segment element
x_4,y_4,z_4	coordinates of the fourth point on the rectangular element
zpf	starting axial location of the vortex cylinder

Table 3

Output Variables

ALP	angle of attack
DP	drag distribution
ETA	location of blade centers
FdP	in plane component of the drag force
FP	thrust distribution
GAMM	bound circulation distribution at the blade centers
LAM	in flow angle
LIP	in plane component of the lift
LP	power coefficient = $\frac{P}{\rho\pi\Omega^3 R^5}$
LT	thrust coefficient = $\frac{T}{\rho\pi\Omega^2 R^4}$
PSI	aximuthal position (in degrees)
	note: the range is from 0 to 360 degrees
R	radial position of the node in the particular PSI
THET	pitch angle
TLIP	thrust component of the lift
TP	In plane component of the force distribution
TWX	
TWY	Same as in Table 2
TWZ	
U	resultant of the induced velocity relative to the blade

WRIV, WTIV, WZIV	Same as in Table 2
WRNV, WTNV, WZNV	Same as in Table 2
WX, WY, WZ	x,y,z component of the induced velocity at the blade centers
XIV, YIV, ZIV	Same as in Table 2
XNV, YNV, ZNV	Same as in Table 2

ORIGINAL PAGE IS
OF POOR QUALITY

Table 4. TYPICAL OUTPUT

```
*** INPUT *** CASE # : 1
NUMBER OF BLADES 2
SIGMA     0.03640
MU     0.60000
 BLADES ARE TWISTED
PITCH ANGLE DISTRIBUTION(DG):  16.95000  15.30000  14.20000  13.65000  12.55000
      11.93000  10.46000   9.81000   9.67000   9.25000   8.70000
       8.40000   8.15000   7.93000   7.71000
PITCH ANGLE DISTRIBUTION(DG):   7.49000   7.22000   7.05000
STALL ANGLE      0.20000
 CD=.0087-0.300*ALPHA**2
ROOT AND TIP VORTICES FACTOR:   1.00000
MAX.NUMBERSOFITERATIONSFORBSIMPL,LOOP1,LOOP2: 50 50 20
CORE SIZE FOR SURFACE ELEMENTS:   0.05000 FOR SEGMENT ELEMENTS:   0.01500
NEAR WAKE DEFINITION:(19, 5),   0.15000   0.25000   0.35000   0.60000
       0.55000   0.65000   0.69000   0.74000   0.76000   0.80000
  0.85000   0.88000   0.90000   0.92000
NEAR WAKE DEFINITION:(**,**),   0.96500   1.00000
INT. WAKE DEFINITION:( 6, 5),   0.15000   0.25000   0.40000   0.75000
       0.95000   1.00000
NW/R  5 NIW  5
ISANGL  0 ISAMET   6
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
SEMI-RIGID WAKE ANALYSIS - CONVERGED AFTER  20 ITERATIONS

ETA     0.175000  0.300000  0.375000  0.450000  0.575000  0.670000  0.715000  0.750000  0.780000
        0.825000  0.865000  0.890000  0.910000  0.930000  0.950000  0.972500  0.992530

WX      0.013975 -0.003451 -0.008621 -0.012453 -0.018959 -0.024983 -0.027912 -0.030474 -0.032887
       -0.037044 -0.041398 -0.044433 -0.046962 -0.049454 -0.051605 -0.053318 -0.053608
WY     -0.010659 -0.007805 -0.006637 -0.005743 -0.004733 -0.003067 -0.002752 -0.002581 -0.002416
       -0.002217 -0.002062 -0.001968 -0.001896 -0.001855 -0.001842 -0.001825 -0.004467
WZ      0.043099  0.054073  0.056901  0.058708  0.061962  0.064459  0.063976  0.064102  0.063815
        0.063605  0.063325  0.063023  0.062707  0.062595  0.063862  0.067574  0.075637
U       0.170229  0.295272  0.375184  0.448778  0.572649  0.645774  0.715603  0.752914  0.780256
        0.825220  0.845850  0.889608  0.911141  0.931036  0.950238  0.973021  0.994673

LAM     0.254295  0.180116  0.152994  0.131247  0.104966  0.095391  0.089372  0.085781  0.082075
        0.077466  0.074053  0.071784  0.070198  0.069020  0.068487  0.070004  0.075999
THET    0.251434  0.257616  0.243057  0.228455  0.204440  0.184431  0.177762  0.171042  0.165285
        0.156663  0.149226  0.144426  0.140326  0.136495  0.132665  0.128369  0.124529
ALPH   -0.077539 -0.077920 -0.090063 -0.077391 -0.097474 -0.091010 -0.088390 -0.085151 -0.093208
       -0.079175 -0.075171 -0.072442 -0.070126 -0.067465 -0.064157 -0.058363 -0.041610
GAMM   -0.001058 -0.005228 -0.007735 -0.010308 -0.012807 -0.013937 -0.014471 -0.014459 -0.014866
       -0.014961 -0.014934 -0.014797 -0.014630 -0.014382 -0.013655 -0.013005 -0.011076

LIP    -0.000180 -0.001544 -0.002902 -0.004491 -0.007336 -0.008320 -0.010147 -0.011053 -0.011599
       -0.012346 -0.012905 -0.013165 -0.013330 -0.013391 -0.013255 -0.012652 -0.011017
TLIP   -0.000174 -0.001519 -0.002864 -0.004453 -0.007292 -0.009278 -0.010106 -0.011012 -0.011560
       -0.012309 -0.012870 -0.013132 -0.013297 -0.013359 -0.013234 -0.012621 -0.010985
FLIP    0.000045  0.000277  0.000442  0.000583  0.000791  0.000832  0.000923  0.000948  0.000951
        0.000955  0.000955  0.000944  0.000935  0.000923  0.000902  0.000885  0.000835

DP      0.000035  0.000054  0.000093  0.000139  0.000226  0.000298  0.000336  0.000344  0.000387
        0.000425  0.000460  0.000480  0.000498  0.000516  0.000528  0.000542  0.000547
TDP     0.000004  0.000010  0.000014  0.000018  0.000024  0.000028  0.000030  0.000031  0.000032
        0.000033  0.000034  0.000034  0.000035  0.000035  0.000036  0.000038  0.000042
FDP     0.000035  0.000053  0.000097  0.000136  0.000223  0.000294  0.000332  0.000343  0.000386
        0.000424  0.000457  0.000479  0.000497  0.000511  0.000527  0.000540  0.000546

TP     -0.000370 -0.001509 -0.002854 -0.004435 -0.007260 -0.009256 -0.010276 -0.010951 -0.011529
       -0.012276 -0.012836 -0.013095 -0.013262 -0.013321 -0.013198 -0.012583 -0.010943
FP      0.000066  0.000333  0.000536  0.000726  0.001006  0.001162  0.001256  0.001311  0.001337
        0.001375  0.001415  0.001425  0.001432  0.001436  0.001435  0.001425  0.001331

          CT=   -0.00379      CP=     0.00034
          CT=
Location of rollup vortices =    0.28756   0.40000   0.58025   0.97286   0.00007  -0.00227  -0.00475   0.01496
    GAMMAS  -0.00345  -0.00571  -0.00819  -0.01042  -0.01316  -0.01432  -0.01487  -0.01510  -0.01531  -0.01546  -0.01547  -0.01539  -0.0
1523  -0.01495  -0.01445  -0.01341   0.00000
Location of rollup vortices =    0.25492   0.45520   0.68918   0.97925   0.00000  -0.00497  -0.00231   0.01547
    GAMMAS  -0.00376  -0.00605  -0.00851  -0.01068  -0.01338  -0.01458  -0.01513  -0.01539  -0.01562  -0.01583  -0.01592  -0.01568  -0.0
1573  -0.01565  -0.01489  -0.01377   0.00000
Location of rollup vortices =    0.28606   0.45553   0.69736   0.97946   0.00003  -0.00487  -0.00256   0.01592
    GAMMAS  -0.00292  -0.00754  -0.00930  -0.01168  -0.01418  -0.01550  -0.01609  -0.01646  -0.01680  -0.01729  -0.01779  -0.01799  -0.0
1746  -0.01765  -0.01679  -0.01537   0.00000
Location of rollup vortices =    0.28280   0.50170   0.77608   0.97959   0.00003  -0.00571  -0.00249   0.01799
    GAMMAS  -0.00316  -0.00795  -0.01015  -0.01158  -0.01427  -0.01564  -0.01627  -0.01673  -0.01714  -0.01784  -0.01862  -0.01898  -0.0
1898  -0.01876  -0.01764  -0.01615   0.00000
Location of rollup vortices =    0.28151   0.57485   0.80935   0.97957   0.00003  -0.00611  -0.00272   0.01899
    GAMMAS  -0.00315  -0.00813  -0.01034  -0.01195  -0.01415  -0.01561  -0.01627  -0.01681  -0.01735  -0.01833  -0.01946  -0.02000  -0.0
```

ORIGINAL PAGE IS
OF POOR QUALITY

[illegible] -0.01974 -0.01964 -0.01696 0.00000

*** MAIN ***; FINAL RESULTS - CONVERGED AFTER 5 ITERATIONS

REL WAKE ANALYSIS

```
      ((BO=0.G14, (BK=0.500)
ETA     0.175000  0.300000  0.375000  0.450000  0.575000  0.670000  0.715000  0.750000  0.780000
        0.825000  0.865000  0.890000  0.910000  0.930000  0.950000  0.972500  0.992530

WKC    -0.004734 -0.004683 -0.005498 -0.008062 -0.016998 -0.025675 -0.030476 -0.034583 -0.038342
       -0.043728 -0.045911 -0.044673 -0.042306 -0.039241 -0.036009 -0.032576 -0.029867
WTC    -0.000097 -0.000657 -0.001166 -0.001511 -0.001680 -0.001627 -0.001525 -0.001395 -0.001224
       -0.000826 -0.000451 -0.000391 -0.000467 -0.000457 -0.001016 -0.001670  0.002471
WIC     0.036206  0.042163  0.046282  0.050915  0.055999  0.056791  0.056127  0.054911  0.053275
        0.049259  0.044155  0.041263  0.040021  0.040532  0.044591  0.050730 -0.030496

U       0.178616  0.302298  0.376686  0.451369  0.574049  0.670781  0.715679  0.750616  0.780596
        0.825646  0.865466  0.890545  0.910413  0.930227  0.950031  0.972355  0.995638
GAC    -0.003162 -0.008133 -0.010336 -0.011947 -0.014150 -0.015616 -0.016265 -0.016813 -0.017336
       -0.018328 -0.019464 -0.019997 -0.020086 -0.019787 -0.018643 -0.016959  0.000000

LAM     0.204118  0.139952  0.123176  0.113043  0.097365  0.084755  0.078506  0.073220  0.068300
        0.059697  0.051030  0.046351  0.043973  0.043536  0.046943  0.052196 -0.030636
ALP    -0.077317 -0.117536 -0.119861 -0.115595 -0.107276 -0.101636 -0.099256 -0.097823 -0.096983
       -0.096966 -0.098196 -0.098575 -0.096351 -0.092899 -0.085732 -0.076173 -0.155163
THE     0.281636  0.257636  0.243017  0.228638  0.206660  0.186631  0.177762  0.171042  0.165283
        0.156643  0.149226  0.144426  0.140326  0.136635  0.132645  0.128369  0.124529

LTP    -0.000565 -0.002459 -0.003896 -0.005593 -0.008151 -0.010471 -0.011641 -0.012620 -0.013331
       -0.015132 -0.016849 -0.017811 -0.018286 -0.018407 -0.017712 -0.016491  0.000000
TLTP   -0.000553 -0.002435 -0.003865 -0.005558 -0.008117 -0.010436 -0.011605 -0.012586 -0.013500
       -0.015105 -0.016827 -0.017791 -0.018268 -0.018389 -0.017692 -0.016468  0.000000
FLTP    0.000114  0.000343  0.000478  0.000603  0.000792  0.000887  0.000915  0.000923  0.000923
        0.000901  0.000859  0.000825  0.000806  0.000802  0.000831  0.000860  0.000000
GP      0.000029  0.000070  0.000110  0.000154  0.000239  0.000316  0.000353  0.000386  0.000415
        0.000465  0.000514  0.000544  0.000563  0.000578  0.000591  0.000582  0.000961
TGP     0.000004  0.000010  0.000013  0.000017  0.000023  0.000022  0.000028  0.000028  0.000028
        0.000029  0.000026  0.000025  0.000025  0.000025  0.000027  0.000030 -0.000029
FGP     0.000019  0.000069  0.000107  0.000153  0.000238  0.000318  0.000352  0.000385  0.000416
        0.000464  0.000513  0.000543  0.000553  0.000577  0.000551  0.000562  0.000960
TP     -0.000549 -0.002425 -0.003851 -0.005543 -0.008089 -0.010497 -0.011577 -0.012558 -0.013471
       -0.015078 -0.016831 -0.017766 -0.018244 -0.018364 -0.017665 -0.016438 -0.000029
FP      0.000114  0.000412  0.000587  0.000761  0.001030  0.001206  0.001265  0.001308  0.001338
        0.001327  0.001373  0.001363  0.001366  0.001379  0.001412  0.001442  0.000960

    CT =-0.455926E-02
    CP = 0.332531E-03
    CI =

    CPI =  0.227613E-03
```

ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

NEE WAKE ANALYSIS

((DO=0.014, CD=1.000)

ETA	0.175000	0.300000	0.375000	0.450000	0.575000	0.670000	0.715000	0.750000	0.790000
	0.825000	0.865000	0.890000	0.910000	0.930000	0.950000	0.972500	0.992530	
WXC	-0.004734	-0.004643	-0.005495	-0.035062	-0.014973	-0.025675	-0.030474	-0.034593	-0.038342
	-0.043725	-0.045911	-0.044673	-0.042306	-0.039241	-0.036009	-0.032576	-0.329847	
WYC	-0.000692	-0.000257	-0.001156	-0.001511	-0.001680	-0.001677	-0.001525	-0.001375	-0.001226
	-0.000825	-0.000461	-0.000321	-0.000467	-0.000657	-0.001016	-0.001470	0.002671	
WZC	0.036206	0.042163	0.046282	0.050915	0.055999	0.056731	0.054127	0.054911	0.053275
	0.049259	0.044155	0.041253	0.040021	0.040532	0.044551	0.050730	-0.030476	
U	0.178516	0.302298	0.376684	0.451369	0.576049	0.670791	0.715579	0.750616	0.780596
	0.825444	0.855665	0.890565	0.910613	0.930227	0.950031	0.972355	0.995615	
GMC	-0.003162	-0.038135	-0.010559	-0.011967	-0.014156	-0.015610	-0.016265	-0.015613	-0.017336
	-0.014328	-0.019454	-0.019999	-0.020086	-0.019797	-0.018643	-0.016959	0.100010	
LAM	0.204113	0.159912	0.123175	0.113063	0.097365	0.084765	0.078506	0.073220	0.068500
	0.059697	0.051010	0.044351	0.043975	0.043586	0.044343	0.052196	-0.030636	
ALP	-0.077317	-0.117516	-0.119861	-0.115595	-0.107274	-0.101636	-0.099256	-0.097833	-0.096983
	-0.094946	-0.098196	-0.094075	-0.096331	-0.092899	-0.085702	-0.074173	-0.155153	
THE	0.231434	0.257436	0.243037	0.220438	0.204640	0.186401	0.177762	0.171042	0.165283
	0.156643	0.149226	0.144426	0.140224	0.136495	0.132645	0.128369	0.124529	
CLP	-0.000565	-0.002459	-0.003694	-0.005393	-0.008151	-0.010471	-0.011641	-0.012620	-0.013531
	-0.015132	-0.016949	-0.017811	-0.018296	-0.018407	-0.017712	-0.016491	0.300010	
TCLP	-0.000553	-0.002435	-0.003565	-0.005358	-0.008112	-0.010434	-0.011605	-0.012536	-0.013500
	-0.015105	-0.016827	-0.017721	-0.018268	-0.018359	-0.017692	-0.016468	0.000010	
FCLP	0.000114	0.000343	0.000478	0.000608	0.000792	0.000857	0.000913	0.000923	0.000923
	0.000903	0.000859	0.000825	0.000804	0.000802	0.000831	0.000860	0.000010	
DP	0.000023	0.000093	0.000147	0.000203	0.000308	0.000399	0.000445	0.000434	0.000520
	0.000581	0.000646	0.000683	0.000705	0.000714	0.000792	0.000522	0.001375	
TDP	0.000005	0.000013	0.000018	0.000023	0.000030	0.000034	0.000035	0.000035	0.000035
	0.000035	0.000035	0.000032	0.000031	0.000031	0.000031	0.000036	-0.000042	
FDP	0.000023	0.000092	0.000146	0.000232	0.000307	0.000378	0.000444	0.000483	0.000519
	0.000580	0.000645	0.000682	0.000703	0.000715	0.000731	0.000281	0.001375	
TP	-0.000548	-0.002422	-0.003847	-0.005335	-0.008082	-0.010400	-0.011570	-0.012551	-0.013464
	-0.015071	-0.016794	-0.017760	-0.018258	-0.018358	-0.017459	-0.016433	-0.000042	
FP	0.000137	0.000435	0.000624	0.000810	0.001099	0.001254	0.001357	0.001436	0.001442
	0.001483	0.001506	0.001597	0.001536	0.001515	0.001532	0.001542	0.001375	

CT =-0.455555E-02
CP = 0.368172E-03
CI =

CPI = 0.22761E-03

FREE WAKE ANALYSIS

(CDO=0.014, CDK=1.500)

ETA	0.175000	0.300000	0.375000	0.450000	0.575000	0.670000	0.715000	0.750010	0.790000
	0.825000	0.865000	0.890000	0.910000	0.930000	0.950000	0.972500	0.992500	
WXC	-0.004734	-0.004443	-0.005494	-0.008082	-0.016998	-0.025475	-0.030474	-0.034593	-0.038342
	-0.043728	-0.045911	-0.044671	-0.042306	-0.039741	-0.036049	-0.032576	-0.029967	
WYC	-0.000092	-0.000457	-0.001166	-0.001531	-0.001480	-0.001227	-0.001525	-0.001375	-0.001226
	-0.000826	-0.000461	-0.000391	-0.000447	-0.000457	-0.001016	-0.001470	0.002471	
WZC	0.016205	0.042165	0.046787	0.050915	0.055999	0.056771	0.056127	0.054911	0.053273
	0.049257	0.044155	0.041263	0.040021	0.040532	0.044581	0.050750	-0.030476	
U	0.178614	0.302298	0.376288	0.451349	0.576049	0.670781	0.715679	0.750616	0.780596
	0.825544	0.865444	0.890565	0.910413	0.930227	0.950031	0.972355	0.995418	
GMC	-0.003162	-0.009133	-0.010339	-0.011967	-0.014150	-0.015610	-0.016265	-0.016813	-0.017336
	-0.018128	-0.019464	-0.019999	-0.020086	-0.019787	-0.019463	-0.016957	0.000010	
LAM	0.204118	0.159912	0.123175	0.112043	0.097345	0.084765	0.079506	0.073220	0.068303
	0.059627	0.051030	0.046151	0.043273	0.041596	0.046743	0.052176	-0.030616	
ALP	-0.077317	-0.117394	-0.119261	-0.115595	-0.107774	-0.101612	-0.099256	-0.107823	-0.076983
	-0.076964	-0.098176	-0.099075	-0.096551	-0.097899	-0.085702	-0.076173	-0.155163	
THE	0.281436	0.257436	0.243037	0.222638	0.204640	0.186601	0.177742	0.171062	0.165283
	0.156643	0.149276	0.144426	0.140326	0.136425	0.132445	0.128349	0.124523	
CIP	-0.000565	-0.002459	-0.003896	-0.005393	-0.008151	-0.010471	-0.011461	-0.012429	-0.013531
	-0.015132	-0.016849	-0.017321	-0.018236	-0.014407	-0.017712	-0.012491	0.000010	
CLIP	-0.000553	-0.002455	-0.003865	-0.005358	-0.008112	-0.010434	-0.011605	-0.012596	-0.013500
	-0.015105	-0.016527	-0.017791	-0.018268	-0.018399	-0.017692	-0.014468	0.000033	
FLIP	0.000114	0.000163	0.000478	0.000608	0.000792	0.000857	0.000913	0.000923	0.000925
	0.000903	0.000859	0.000825	0.000806	0.000802	0.000831	0.000860	0.000010	
DP	0.000027	0.000116	0.000184	0.000255	0.000378	0.000484	0.000537	0.000592	0.000626
	0.000698	0.000777	0.000822	0.000843	0.000850	0.000823	0.000782	0.001810	
TDP	0.000015	0.000016	0.000023	0.000029	0.000037	0.000041	0.000042	0.000043	0.000043
	0.000042	0.000040	0.000033	0.000037	0.000037	0.000039	0.000041	-0.000055	
FDP	0.000026	0.000114	0.000182	0.000251	0.000376	0.000492	0.000536	0.000591	0.000623
	0.000597	0.000776	0.000821	0.000843	0.000849	0.000822	0.000781	0.001810	
CP	-0.000548	-0.002419	-0.003842	-0.005330	-0.008045	-0.010373	-0.011543	-0.012544	-0.013437
	-0.015064	-0.016758	-0.017753	-0.018231	-0.018352	-0.017653	-0.014427	-0.000035	
FP	0.000141	0.000457	0.000661	0.000859	0.001149	0.001348	0.001448	0.001514	0.001546
	0.001600	0.001636	0.001646	0.001647	0.001651	0.001653	0.001642	0.001810	

CT =-0.455528E-02
CP = 0.398018E-03
CI =

CPI = 0.227618E-03

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS OF POOR QUALITY

NEE WAKE ANALYSIS

(CDU=0.014, CDA=2.000)

ETA	0.175000	0.300000	0.375000	0.450000	0.575000	0.670000	0.715000	0.750000	0.790000
	0.825000	0.865000	0.890000	0.910000	0.930000	0.950000	0.972500	0.992510	
UTC	-0.005734	-0.006455	-0.005494	-0.008052	-0.016999	-0.025675	-0.030674	-0.036545	-0.038342
	-0.043725	-0.045911	-0.044673	-0.042306	-0.039241	-0.036009	-0.032576	-0.029867	
VTC	-0.000072	-0.000657	-0.001165	-0.001511	-0.001690	-0.001627	-0.001525	-0.001375	-0.001226
	-0.000826	-0.000461	-0.000391	-0.000467	-0.000657	-0.001016	-0.001470	0.002671	
WZC	0.038206	0.042153	0.045232	0.050915	0.055999	0.056791	0.056127	0.054911	0.053275
	0.049259	0.044155	0.041265	0.040021	0.040532	0.044581	0.050730	-0.030496	
U	0.178616	0.302298	0.376655	0.451369	0.574049	0.670781	0.715679	0.750616	0.780596
	0.825644	0.865656	0.890565	0.910615	0.930227	0.950031	0.972355	0.995618	
UNC	-0.003152	-0.008133	-0.010334	-0.011767	-0.014150	-0.015616	-0.016265	-0.016813	-0.017336
	-0.018328	-0.019444	-0.019999	-0.020086	-0.019757	-0.018443	-0.016959	0.000030	
LAM	0.164118	0.139932	0.123175	0.113043	0.097565	0.084765	0.078506	0.073220	0.048300
	0.059597	0.051030	0.046551	0.043973	0.043586	0.046943	0.052192	-0.030634	
ALP	-0.077317	-0.117596	-0.119861	-0.115595	-0.107274	-0.101636	-0.099256	-0.097823	-0.096983
	-0.095966	-0.098196	-0.098075	-0.096351	-0.092899	-0.085732	-0.076173	-0.155153	
THE	0.281434	0.257436	0.243037	0.222638	0.204640	0.184401	0.177762	0.171042	0.165283
	0.156643	0.149226	0.144426	0.140324	0.136495	0.132645	0.128369	0.124529	
CIP	-0.000563	-0.002459	-0.003896	-0.005393	-0.008151	-0.010471	-0.011641	-0.012620	-0.013531
	-0.015132	-0.016849	-0.017811	-0.018256	-0.018407	-0.017712	-0.016491	0.000000	
TLIP	-0.000553	-0.002435	-0.003865	-0.005359	-0.008112	-0.010434	-0.011605	-0.012536	-0.013500
	-0.015105	-0.016327	-0.017721	-0.018268	-0.018389	-0.017692	-0.016463	0.000070	
FLIP	0.000114	0.000363	0.000478	0.000608	0.000792	0.000857	0.000913	0.000923	0.000923
	0.000901	0.000859	0.000825	0.000694	0.000802	0.000831	0.000860	0.000090	
DP	0.000030	0.000139	0.000221	0.000302	0.000448	0.000568	0.000629	0.000640	0.000729
	0.000815	0.000909	0.000961	0.000954	0.000956	0.000944	0.000832	0.002245	
TCP	0.000006	0.000019	0.000027	0.000034	0.000044	0.000042	0.000049	0.000050	0.000050
	0.000049	0.000046	0.000045	0.000043	0.000043	0.000044	0.000046	-0.000069	
FDP	0.000030	0.000137	0.000219	0.000300	0.000446	0.000566	0.000627	0.000679	0.000727
	0.000513	0.000908	0.000960	0.000983	0.000985	0.000743	0.000881	0.002244	
TP	-0.000547	-0.002415	-0.003838	-0.005324	-0.008069	-0.010386	-0.011556	-0.012537	-0.013450
	-0.015057	-0.016791	-0.017767	-0.018225	-0.018346	-0.017648	-0.016422	-0.000059	
FP	0.000144	0.000480	0.000694	0.000909	0.001238	0.001453	0.001540	0.001602	0.001650
	0.001715	0.001767	0.001785	0.001787	0.001787	0.001774	0.001741	0.002246	

CT =-0.455018E-02
CP = 0.42785E-03
CT =

CPI = 0.22763E-03

```
ETAW    0.100000  0.250000  0.350000  0.400000  0.500000  0.650000  0.690000  0.740000  0.760000
        0.800000  0.850000  0.880000  0.900000  0.920000  0.940000  0.960000  0.985000  1.000000
  1
XNV     0.281506  0.524854  0.809347  0.979574
YNV     0.000000  0.000000  0.000000  0.000000
ZNV     0.000000  0.000000  0.000000  0.000000

WXNV   -0.004050 -0.011440 -0.033718 -0.032586
WYNV   -0.000580 -0.001593 -0.001024  0.000058
WZNV    0.038008  0.050597  0.052062  0.025531

PSI        0.000     0.000     0.000     0.000
R       0.281506  0.524854  0.809347  0.979574

  2
XNV     0.276556  0.475960  0.689869  0.959223
YNV     0.048373  0.083460  0.121319  0.169179
ZNV     0.008062  0.010380  0.011077  0.009695

WXNV   -0.004582 -0.008678 -0.023091 -0.030962
WYNV   -0.003863 -0.003456 -0.002774  0.000419
WZNV    0.054380  0.068353  0.074873  0.085539

PSI        9.921     9.946     9.974    10.002
R       0.280755  0.483203  0.700455  0.974028

  3
XNV     0.262386  0.451876  0.654266  0.910200
YNV     0.094887  0.163301  0.237301  0.331401
ZNV     0.016806  0.021473  0.023449  0.024929

WXNV   -0.004606 -0.008627 -0.022615 -0.030625
WYNV   -0.002298 -0.002585 -0.002324  0.000351
WZNV    0.045515  0.058758  0.066899  0.089063

PSI       19.912    19.878    19.936    20.005
R       0.279957  0.480595  0.695071  0.968454

  4
XNV     0.242380  0.414521  0.599541  0.834180
YNV     0.138505  0.237693  0.344762  0.481803
ZNV     0.024522  0.031447  0.034963  0.040017

WXNV   -0.004540 -0.008549 -0.022276 -0.030476
WYNV   -0.001447 -0.002114 -0.002282  0.000290
WZNV    0.042599  0.055542  0.065045  0.003834

PSI       29.745    29.827    29.901    30.010
R       0.279162  0.477872  0.691599  0.963323

  5
XNV     0.214171  0.365258  0.527564  0.733726
YNV     0.177835  0.304143  0.440567  0.615939
ZNV     0.031782  0.040849  0.046247  0.051508

WXNV   -0.004492 -0.008275 -0.021973 -0.030736
WYNV   -0.000853 -0.001506 -0.002387  0.000221
WZNV    0.040605  0.052767  0.064260  0.047841
```

PSI	39.704	39.734	39.855	40.015		
R	0.278579	0.475307	0.687339	0.957982		
ETAI	0.100000	0.250000	0.600000	0.850000	0.950000	1.000000
1						
XIV	0.214171	0.365258	0.527564	0.733726		
YIV	0.177835	0.306153	0.440557	0.615935		
ZIV	0.031782	0.040829	0.046247	0.051505		
WXIV	-0.004482	0.008275	-0.021973	-0.030734		
WYIV	-0.000853	-0.011806	-0.002357	0.000221		
WZIV	0.040605	0.052767	0.064260	0.047841		
PSI	39.706	39.784	39.865	40.013		
R	0.278579	0.475337	0.687339	0.957982		
2						
XIV	0.139836	0.237250	0.341679	0.473358		
YIV	0.239069	0.406179	0.585760	0.820369		
ZIV	0.045649	0.055739	0.064677	0.062817		
WXIV	-0.003750	-0.008055	-0.021331	-0.031419		
WYIV	0.000106	-0.001557	-0.002762	0.000097		
WZIV	0.038844	0.049450	0.064263	0.014955		
PSI	59.674	59.711	59.753	60.015		
R	0.276945	0.470393	0.678987	0.947139		
3						
XIV	0.049403	0.083721	0.119954	0.162320		
YIV	0.271216	0.458102	0.660270	0.921687		
ZIV	0.039095	0.075920	0.091177	0.068859		
WXIV	-0.003548	-0.007951	-0.021015	-0.032074		
WYIV	0.000075	-0.001454	-0.002650	-0.000075		
WZIV	0.038198	0.048987	0.064645	0.017836		
PSI	79.577	79.643	79.703	80.014		
R	0.275678	0.465690	0.671078	0.936068		
4						
XIV	-0.046160	-0.076740	-0.110869	-0.160773		
YIV	0.270552	0.454296	0.654149	0.910702		
ZIV	0.072357	0.092978	0.113831	0.075275		
WXIV	-0.003292	-0.007776	-0.020472	-0.032656		
WYIV	0.000050	-0.001440	-0.002687	-0.000144		
WZIV	0.037786	0.048748	0.065147	0.018750		
PSI	99.681	99.578	99.619	100.012		
R	0.274491	0.461225	0.663475	0.924784		
5						
XIV	-0.135364	-0.225094	-0.323389	-0.456774		
YIV	0.237552	0.397718	0.570852	0.790896		
ZIV	0.095459	0.109931	0.136634	0.081967		
WXIV	-0.002925	-0.007487	-0.019841	-0.033125		
WYIV	-0.000198	-0.001591	-0.002779	-0.000185		
WZIV	0.037282	0.048395	0.065507	0.019595		

PSI	119.676	119.509	119.532	120.009
R	0.273412	0.456998	0.656089	0.913323
6				
XIV	-0.207593	-0.344051	-0.493014	-0.690817
YIV	0.176511	0.294723	0.422049	0.579592
ZIV	0.098325	0.126677	0.159480	0.088905
WXIV	-0.002393	-0.006963	-0.018908	-0.033328
WYIV	-0.001160	-0.002381	-0.003259	-0.000238
WZIV	0.036434	0.047564	0.065394	0.020159
PSI	139.627	139.417	139.435	140.004
R	0.272490	0.453034	0.648990	0.901751
7				
XIV	-0.254393	-0.420105	-0.600816	-0.836539
YIV	0.095502	0.159392	0.227234	0.304648
ZIV	0.110911	0.143088	0.182171	0.095846
WXIV	-0.001995	-0.006388	-0.017347	-0.032558
WYIV	-0.004399	-0.005968	-0.006073	-0.001014
WZIV	0.035660	0.046465	0.064615	0.019609
PSI	159.424	159.223	159.284	159.990
R	0.271729	0.449327	0.642351	0.890295
8				
XIV	-0.270648	-0.445530	-0.635870	-0.880252
YIV	0.004975	0.009460	0.011470	0.004047
ZIV	0.124031	0.160023	0.205654	0.104743
WXIV	-0.003975	-0.009197	-0.017482	-0.025769
WYIV	-0.008605	-0.012765	-0.013202	-0.021463
WZIV	0.059493	0.050564	0.069937	0.051369
PSI	178.948	178.794	178.967	179.737
R	0.270694	0.445631	0.535974	0.890762
9				
XIV	-0.255100	-0.418449	-0.595510	-0.822961
YIV	-0.055171	-0.135531	-0.201054	-0.290361
ZIV	0.138421	0.178423	0.231039	0.118066
WXIV	-0.008126	-0.012398	-0.017930	-0.017565
WYIV	-0.006567	-0.006910	-0.007316	-0.002423
WZIV	0.042954	0.054559	0.075532	0.044962
PSI	198.464	198.318	198.627	199.466
R	0.258963	0.440733	0.629432	0.872849
10				
XIV	-0.209579	-0.343175	-0.499150	-0.569692
YIV	-0.165150	-0.268417	-0.387322	-0.550720
ZIV	0.151156	0.197514	0.257342	0.133789
WXIV	-0.006070	-0.012078	-0.016641	-0.015786
WYIV	-0.001506	-0.003320	-0.014400	-0.000452
WZIV	0.042617	0.054525	0.075208	0.045128

ORIGINAL PAGE IS
OF POOR QUALITY

PSI	219.239	214.073	218.431	219.433
R	0.266829	0.435926	0.623144	0.867052
11				
XIV	-0.137659	-0.229975	-0.324373	-0.435418
YIV	-0.224974	-0.355576	-0.525116	-0.741949
ZIV	0.148151	0.216492	0.285579	0.169440
WRIV	-0.005660	-0.011470	-0.015702	-0.014447
WTIV	-0.000351	-0.002177	-0.003553	-0.000506
WZIV	0.042156	0.054215	0.075117	0.044542
PSI	235.170	237.940	238.297	239.422
R	0.264797	0.431365	0.617223	0.861779
12				
XIV	-0.055951	-0.089963	-0.125283	-0.157455
YIV	-0.257314	-0.417502	-0.599624	-0.842286
ZIV	0.137826	0.235498	0.309836	0.164778
WRIV	-0.005244	-0.010967	-0.015043	-0.013997
WTIV	0.000067	-0.001786	-1.003255	-0.000314
WZIV	0.041926	0.054161	0.075324	0.343340
PSI	258.159	257.844	258.180	259.413
R	0.262909	0.427051	0.611595	0.856874
13				
XIV	0.037098	0.057091	0.085080	0.139240
YIV	-0.258520	-0.419198	-0.600187	-0.840436
ZIV	0.197445	0.254332	0.336187	0.179689
WRIV	-0.004821	-0.010470	-0.014479	-0.013646
WTIV	0.000086	-0.001734	-0.003197	-0.000265
WZIV	0.041833	0.054255	0.075655	0.042095
PSI	278.165	277.757	278.069	279.406
R	0.261167	0.423058	0.605188	0.852090
14				
XIV	0.122486	0.194652	0.281725	0.415962
YIV	-0.228858	-0.371358	-0.530871	-0.738251
ZIV	0.212059	0.273296	0.362654	0.194160
WRIV	-0.004381	-0.009957	-0.013584	-0.013518
WTIV	-0.000304	-0.002027	-0.003371	-0.000252
WZIV	0.041784	0.054339	0.075991	0.040815
PSI	298.157	297.663	297.955	299.400
R	0.259574	0.419260	0.600995	0.847372
15				
XIV	0.192106	0.306635	0.441648	0.639765
YIV	-0.172398	-0.290650	-0.400110	-0.548519
ZIV	0.226634	0.292396	0.359228	0.208154
WRIV	-0.004035	-0.009468	-0.013234	-0.013293
WTIV	-0.001295	-0.002999	-0.003979	-0.000406
WZIV	0.041841	0.054530	0.076266	0.039365

ORIGINAL PAGE IS
OF POOR QUALITY

PSI	355.096	357.559	357.826	359.397
G	0.258118	0.415715	0.595937	0.842717
16				
XIV	0.237822	0.330492	0.546481	0.794515
YIV	-0.094430	-0.158775	-0.224635	-0.295430
ZIV	0.241341	0.311431	0.415905	0.221606
WXIV	-0.004701	-0.009767	-0.011449	-0.012170
WYIV	-0.007857	-0.006563	-0.005264	-0.011809
WZIV	0.067626	0.055064	0.074580	0.057715
PSI	357.916	357.356	357.663	359.366
G	0.256695	0.412237	0.591036	0.838778
17				
XIV	0.234456	0.408768	0.585746	0.834745
YIV	-0.010330	-0.070768	-0.025978	-0.290951
ZIV	0.256421	0.338914	0.447476	0.235459
WXIV	-0.005210	-0.139772	-0.012395	-0.010759
WYIV	-0.005767	-0.035475	-0.006029	-0.008316
WZIV	0.063975	0.055552	0.075807	0.041459
PSI	357.480	357.959	357.442	358.249
G	0.255065	0.403796	0.586322	0.836317
18				
XIV	0.261475	0.347760	0.555459	0.784989
YIV	0.075773	0.117273	0.172599	0.272141
ZIV	0.273028	0.330905	0.469674	0.250878
WXIV	-0.008265	-0.010445	-0.012359	-0.009363
WYIV	-1.007856	-0.004700	-0.005064	-0.002655
WZIV	0.065644	0.057959	0.076731	0.056686
PSI	17.422	14.821	17.262	19.121
G	0.253084	0.403392	0.581656	0.830824
19				
XIV	0.199687	0.322062	0.460201	0.462606
YIV	0.151866	0.239314	0.360772	0.521820
ZIV	0.248606	0.371257	0.496315	0.267071
WXIV	-0.006532	-0.010519	-0.012362	-0.008116
WYIV	-0.001608	-0.001111	-0.001686	-0.022890
WZIV	0.066134	0.055547	0.077062	0.044097
PSI	57.754	56.617	57.104	59.078
G	0.250876	0.431255	0.577021	0.827790
20				
XIV	0.114766	0.219470	0.311925	0.424175
YIV	1.208972	0.333145	0.480026	0.707614
ZIV	0.304156	0.391810	0.525325	0.283016
WXIV	-0.006101	-0.010175	-0.012235	-0.007898
WYIV	-0.000609	-0.002098	-0.002857	-0.030636
WZIV	0.064429	0.059083	0.077639	0.045362

PSI	57.192	56.491	56.776	59.063
R	0.269659	0.377491	0.572467	0.825313
21				
X1Y	0.054756	0.022727	0.129875	0.156765
Y1Y	0.260327	0.332755	0.553294	0.507733
Z1Y	1.370415	0.417522	0.550573	1.298779
WX1Y	-0.015482	-0.002719	-0.017051	-0.008705
WY1Y	0.000073	-0.031571	-0.032455	-0.000537
WZ1Y	0.064777	0.357592	0.079542	0.064805
PSI	77.164	75.492	76.847	77.066
R	0.268556	0.373927	0.569017	0.822235
22				
X1Y	-0.135556	-0.047917	-0.068711	-0.129426
Y1Y	0.267657	0.351011	0.555677	0.825177
Z1Y	1.512743	0.433620	0.579193	1.514519
WX1Y	-0.005471	-0.012207	-0.011832	-0.009657
WY1Y	0.000317	-0.031531	-0.007336	-0.130532
WZ1Y	0.047073	0.046157	0.077556	0.044487
PSI	97.176	96.278	96.779	99.031
R	0.266605	0.370357	0.563615	0.819247
23				
X1Y	-0.110976	-0.170266	-0.251323	-0.395731
Y1Y	0.217604	0.347226	0.499651	0.711732
Z1Y	1.553773	0.456527	0.606156	1.529696
WX1Y	-0.005025	-0.005779	-0.011533	-0.009144
WY1Y	-0.000237	-0.001805	-0.002477	-0.000616
WZ1Y	0.047497	0.050776	0.050662	0.043757
PSI	117.177	115.701	116.798	119.019
R	0.262207	0.387786	0.559357	0.816175
24				
X1Y	-0.176677	-0.276637	-0.403417	-0.615546
Y1Y	0.164059	0.246772	0.391615	0.533357
Z1Y	1.563957	0.475951	0.634560	0.344855
WX1Y	-0.004265	-0.009421	-0.011263	-0.009621
WY1Y	-0.000357	-0.002458	-0.002924	-0.000950
WZ1Y	0.047722	0.053406	0.041746	0.043153
PSI	137.123	136.072	136.606	139.000
R	0.261076	0.393930	0.555183	0.812962
25				
X1Y	-0.220369	-0.367727	-0.503255	-0.755742
Y1Y	0.043697	0.155518	0.219859	0.290680
Z1Y	0.396277	0.497317	0.665165	0.359864
WX1Y	-0.005170	-0.005678	-0.011379	-0.009239
WY1Y	-0.001785	-0.003346	-0.003635	-0.002122
WZ1Y	0.049665	0.062011	0.082676	0.062866

PSI	157.010	155.933	156.482	158.962
B	0.239362	0.350836	0.551030	0.809716
26				
XIV	-0.237072	-0.376435	-0.565512	-0.807206
YIV	0.013062	0.928077	0.036932	0.015556
ZIV	0.403958	0.519115	0.691866	0.375325
WXIV	-0.006009	-0.009721	-0.012633	-0.036377
WYIV	-0.002157	-0.003835	-0.003956	-0.003235
WZIV	0.049629	0.062653	0.081870	0.045741
PSI	176.847	175.735	176.337	178.897
B	0.237632	0.377481	0.546629	0.807356
27				
XIV	-0.225316	-0.340035	-0.520253	-0.745556
YIV	-0.067526	-0.100032	-0.151137	-0.240328
ZIV	0.421376	0.540981	0.720315	0.391782
WXIV	-0.004857	-0.011036	-0.013975	0.000677
WYIV	-0.001767	-0.003273	-0.003155	-0.002615
WZIV	0.050296	0.062810	0.081266	0.048548
PSI	196.686	195.557	196.200	198.827
B	0.235215	0.373649	0.561759	0.806716
28				
XIV	-0.186955	-0.301326	-0.433629	-0.629365
YIV	-0.138700	-0.213985	-0.316169	-0.505659
ZIV	0.639050	0.563017	0.748826	0.402641
WXIV	-0.007268	-0.011683	-0.014001	0.000315
WYIV	-0.000908	-0.002265	-0.002203	-0.001172
WZIV	0.050985	0.063628	0.082101	0.048047
PSI	216.572	215.351	216.096	218.785
B	0.232787	0.367576	0.536642	0.806776
29				
XIV	-0.127028	-0.208163	-0.297135	-0.619639
YIV	-0.192110	-0.300360	-0.449270	-0.690121
ZIV	0.456956	0.555287	0.777721	0.425769
WXIV	-0.007153	-0.011249	-0.013293	0.000786
WYIV	-0.000151	-0.001662	-0.001570	-0.000765
WZIV	0.051605	0.064172	0.083668	0.047228
PSI	236.527	235.275	236.022	238.759
B	0.230309	0.365627	0.531633	0.807171
30				
XIV	-0.053020	-0.092306	-0.127726	-0.157529
YIV	-0.221567	-0.369638	-0.531250	-0.791578
ZIV	0.475077	0.607832	0.807118	0.441636
WXIV	-0.006773	-0.010693	-0.012257	-0.000637
WYIV	0.000292	-0.011002	-0.001225	-0.000611
WZIV	0.052220	0.065001	0.084967	0.046536

PSI	256.333	255.296	255.967	258.742
W	0.227916	0.361423	0.526940	0.807110
31				
XIV	0.025785	0.032097	0.053903	0.122396
YIV	-0.224190	-0.356205	-0.519930	-0.797356
ZIV	0.493417	0.630676	0.837042	0.457773
WXIV	-0.006296	-0.010003	-0.011160	-0.001750
WYIV	0.000371	-0.000869	-0.001113	-0.000574
WZIV	0.052863	0.065875	0.064695	0.045934
PSI	276.562	275.150	275.920	278.728
W	0.225668	0.357649	0.522697	0.806695
32				
XIV	0.100044	0.150185	0.226402	0.387142
YIV	-0.199963	-0.320707	-0.466825	-0.706795
ZIV	0.511989	0.655625	0.867499	0.473713
WXIV	-0.005855	-0.009301	-0.010013	-0.002974
WYIV	0.000097	-0.001035	-0.001202	-0.000665
WZIV	0.055548	0.066766	0.089005	0.045397
PSI	296.583	295.096	295.874	298.715
W	0.223576	0.354130	0.519829	0.805975
33				
XIV	0.160903	0.248168	0.369535	0.606395
YIV	-0.152343	-0.246001	-0.359170	-0.551148
ZIV	0.550811	0.677279	0.898467	0.489485
WXIV	-0.005712	-0.008731	-0.009952	-0.004264
WYIV	-0.000469	-0.001455	-0.001457	-0.001018
WZIV	0.054290	0.067620	0.099425	0.044970
PSI	316.566	315.021	315.820	318.692
W	0.221591	0.350946	0.515561	0.804619
34				
XIV	0.201336	0.316918	0.467035	0.747826
YIV	-0.087371	-0.167406	-0.210347	-0.292266
ZIV	0.569902	0.701004	0.929878	0.505185
WXIV	-0.006083	-0.008492	-0.009230	-0.005625
WYIV	-0.001146	-0.001969	-0.001716	-0.001876
WZIV	0.055131	0.068339	0.090544	0.044986
PSI	336.494	334.918	335.755	338.636
W	0.219554	0.347729	0.512218	0.802902
35				
XIV	0.218957	0.345168	0.507841	0.800414
YIV	-0.013751	-0.031293	-0.038352	-0.019584
ZIV	0.569326	0.724472	0.961591	0.521152
WXIV	-0.007105	-0.008639	-0.007815	-0.007323
WYIV	-0.001557	-0.002257	-0.001805	-0.002677
WZIV	0.056125	0.065997	0.091157	0.044497

```
PSI     356.373     354.791     355.683     358.600
B       0.217292    0.344592    0.509287    0.300654

INFLUENCE COEFF.   IW(I,K)=GAMMA(K)*W(I)

IWY     0.568606749 -0.093537215 -0.050348986 -0.019964027 -0.009013615 -0.003734935 -0.001103088 -0.000886546 -0.000714164
       -0.000656500 -0.000467581 -0.000213640 -0.000269306 -0.000704965 -0.000216196  0.000091479  0.000000000
IWZ    -1.366273001  0.353195466  0.193741136  0.117237560  0.072573210  0.024677337  0.013169094  0.003065861  0.009465393
        0.006601696  0.006443126  0.003462644  0.002812554  0.003383205  0.004641177  0.002493777  0.000000000


IWY    -0.047812097  0.303707529 -0.066164381 -0.064849398 -0.024294740 -0.008849310 -0.001755457 -0.003401629 -0.000691758
       -0.001354287  0.001255620 -0.002703405 -0.000545029 -0.000390584 -0.002789981  0.002590506  0.000000000
IWZ     0.461928900 -1.511643654  0.129004085  0.322190339  0.169619333  0.055517729  0.023290087  0.014841024  0.015503963
        0.015051019  0.009996817  0.005371500  0.004265748  0.005186956  0.006055519  0.003768918  0.000000000


IWY    -0.043586463 -0.046474278  0.304168055 -0.073210629 -0.046622925 -0.014722767 -0.002317538 -0.006130749 -0.000492163
       -0.001993938  0.000556265 -0.002604252 -0.000784994 -0.000549322 -0.001071160  0.000661584  0.000000000
IWZ     0.328626360  0.358950685 -1.959509969  0.490523696  0.343296535  0.098348597  0.035135301  0.022825117  0.022184321
        0.020974961  0.013875832  0.007513858  0.005910140  0.006750792  0.007856546  0.005099057  0.000000000


IWY    -0.016995657 -0.018537276 -0.002109260  0.193680469 -0.040226209 -0.026201167 -0.005724560 -0.067247626 -0.001844027
       -0.002925690 -0.000277262 -0.002638288 -0.001021859 -0.000773670 -0.000871789  0.000167620  0.000000000
IWZ     0.149978606  0.263921607  0.201075731 -1.565647379  0.329766251  0.198178727  0.056677614  0.036845972  0.033237529
        0.030436996  0.018880499  0.010808371  0.008393027  0.009204759  0.010265071  0.006987831  0.000000000


IWY    -0.005256967 -0.010065626 -0.014695217 -0.007279602  0.150052511 -0.004671956 -0.015126603 -0.017647668 -0.005015779
       -0.004970503 -0.004669775 -0.001745626 -0.001845554 -0.001374843 -0.001679870 -0.000253467  0.000000000
IWZ     0.060420352  0.074802727  0.179819902  0.175181366 -1.255546591 -0.030191816  0.176102722  0.105331916  0.077787113
        0.045179220  0.040249029  0.021756264  0.016191714  0.016605495  0.017570071  0.012534491  0.000000000


IWY    -0.002593020 -0.004666761 -0.006835437 -0.020142578 -0.050185224  0.171197275  0.005021658 -0.031534787 -0.015879766
       -0.011297224 -0.007703150 -0.003306979 -0.002879709 -0.002240182 -0.002283322 -0.001020351  0.000000000
IWZ     0.035795336  0.039562524  0.077139278  0.263053532  0.397776503 -2.009474112 -0.065028227  0.359585054  0.212374656
        0.151585776  0.086099776  0.062674479  0.036348070  0.029103720  0.029237559  0.021223414  0.000000000


IWY    -0.001924609 -0.003603277 -0.004812134 -0.012106490 -0.028225036 -0.002458817  0.166639264  0.022167247 -0.029146905
       -0.018937168 -0.011049834 -0.004834195 -0.003731855 -0.002975839 -0.002894696 -0.001654680  0.000000000
IWZ     0.028817388  0.030726671  0.058777225  0.147905160  0.365441151  0.252565745 -1.924621075 -0.355505615  0.405515177
        0.268781215  0.134715834  0.064344072  0.044093397  0.040496423  0.037340998  0.026294299  0.000000000


IWY    -0.001567131 -0.002676683 -0.003762179 -0.008732225 -0.017273565 -0.038342302  0.008254785  0.157880723  0.009184476
       -0.028397364 -0.016024793 -0.004824621 -0.004946263 -0.003645243 -0.003607720 -0.002362119  0.000000000
IWZ     0.024613331  0.025648916  0.047790162  0.108444397  0.224231616  0.610291161 -0.028515000 -2.230792791 -0.140163699
        0.421245111  0.213785721  0.094951567  0.062340643  0.054838140  0.051487381  0.036243249  0.000000000
```

TWY	-0.001309176	-0.002179797	-0.003167529	-0.006304168	-0.012122597	-0.024885559	-0.024456482	0.027719657	0.155289580
	0.003225692	-0.023267076	-0.009752026	-0.006544761	-0.005028005	-0.004514880	-0.001202966	0.000000000	
TWZ	0.021650214	0.022197200	0.040520555	0.026256305	0.158226117	0.394985761	0.412481505	-0.317979515	-2.014711767
	-0.032784186	0.352620114	0.141964832	0.088651448	0.074377618	0.067267380	0.045544806	0.000000000	
TWY	-0.001024763	-0.001541094	-0.002638372	-0.004784698	-0.007676724	-0.013391113	-0.011149903	-0.022499354	0.012509128
	0.130047029	0.016216675	-0.018184673	-0.011424756	-0.008265909	-0.006881776	-0.005085796	0.000000000	
TWZ	0.016044579	0.019143370	0.032214216	0.063990053	0.103398144	0.216433056	0.196512116	0.335482039	-0.130750625
	-1.991959691	-0.278280430	0.284664885	0.171494184	0.131224098	0.109961818	0.068696176	0.000000000	
TWY	-0.000503368	-0.001260119	-0.002102961	-0.003666444	-0.005735515	-0.008664782	-0.005822460	-0.011650114	-0.021077003
	0.005014673	0.125971611	0.032931454	-0.016460882	-0.014463026	-0.011386077	-0.007826582	0.000000000	
TWZ	0.015493267	0.015321516	0.026753569	0.050678599	0.075758507	0.142719628	0.110154134	0.199453973	0.369891681
	-0.000418556	-2.031364968	-0.560779676	0.263119832	0.244324099	0.193595552	0.105962803	0.000000000	
TWY	-0.000707047	-0.001094153	-0.001849704	-0.003134557	-0.004802777	-0.006767324	-0.004370356	-0.007996287	-0.014213283
	-0.023289843	0.049567252	0.113397093	0.035437675	-0.013205474	-0.016084939	-0.010301065	0.000000000	
TWZ	0.014141034	0.013860066	0.023978616	0.044331057	0.063608497	0.116584868	0.081581465	0.139165917	0.248731395
	0.447563923	-0.270818360	-1.914338619	-0.623062299	0.228365839	0.283190083	0.143812425	0.000000000	
TWY	-0.000639357	-0.000981339	-0.001655675	-0.002799505	-0.004213822	-0.005629793	-0.003617060	-0.006079717	-0.010207766
	-0.018487868	-0.021348439	0.035919454	0.102687772	0.034906895	-0.014263515	-0.011986547	0.000000000	
TWZ	0.013171776	0.012821438	0.022040603	0.040057153	0.056175875	0.097775306	0.065956014	0.108186995	0.180066328
	0.350624800	0.415943887	-0.567250870	-1.767599777	-0.620155508	0.256984562	0.170970274	0.000000000	
TWY	-0.000569970	-0.000896624	-0.001498253	-0.002490427	-0.003712856	-0.004672370	-0.003259376	-0.004617568	-0.007559424
	-0.017180211	-0.019585188	-0.013468097	0.036625989	0.100205939	0.033373813	-0.005118182	0.000000000	
TWZ	0.012288987	0.011892628	0.020311995	0.036348659	0.049835626	0.084477235	0.054403913	0.086619716	0.134568578
	0.250014104	0.373690453	0.291881625	-0.606885547	-1.788189619	-0.597143412	0.044356576	0.000000000	
TWY	-0.000497916	-0.000932418	-0.001357650	-0.002229140	-0.003286685	-0.004017874	-0.002754630	-0.003682606	-0.005755546
	-0.009605065	-0.013933987	-0.015017678	-0.010942495	0.035579436	0.102407228	0.057564971	0.000000000	
TWZ	0.011485109	0.011045592	0.018771004	0.033412222	0.044506164	0.073753175	0.045639794	0.070603686	0.103851547
	0.181743131	0.266291726	0.306278296	0.244543130	-0.610194951	-1.857258275	-1.112339929	0.000000000	
TWY	-0.000446316	-0.000744965	-0.001226734	-0.001967065	-0.002885901	-0.003430030	-0.002324404	-0.002914310	-0.004348820
	-0.007049970	-0.009331068	-0.010313087	-0.012595924	-0.012405912	0.033137802	0.198751330	0.000000000	
TWZ	0.010662760	0.010189104	0.017215205	0.029959266	0.039448737	0.063995519	0.038078862	0.057540474	0.08015. 16
	0.132079121	0.182226322	0.211298820	0.262331760	0.234946893	-0.577032700	-3.733951002	0.000000000	
TWY	-0.000303209	-0.000662659	-0.001079885	-0.001697916	-0.002436237	-0.002819586	-0.001832565	-0.002173147	-0.003075448
	-0.004640312	-0.005599485	-0.005779814	-0.006850962	-0.007811106	-0.014317995	-0.010356945	0.000000000	

TWZ	0.009762031	0.039283097	0.015553985	0.026548030	0.034044220	0.053749623	0.030382130	0.044820437	0.057964413
	0.088355093	0.112218907	0.120580710	0.147024427	0.203209564	0.292982865	0.623957440	0.000000000	

Table 5. FLOW CHART

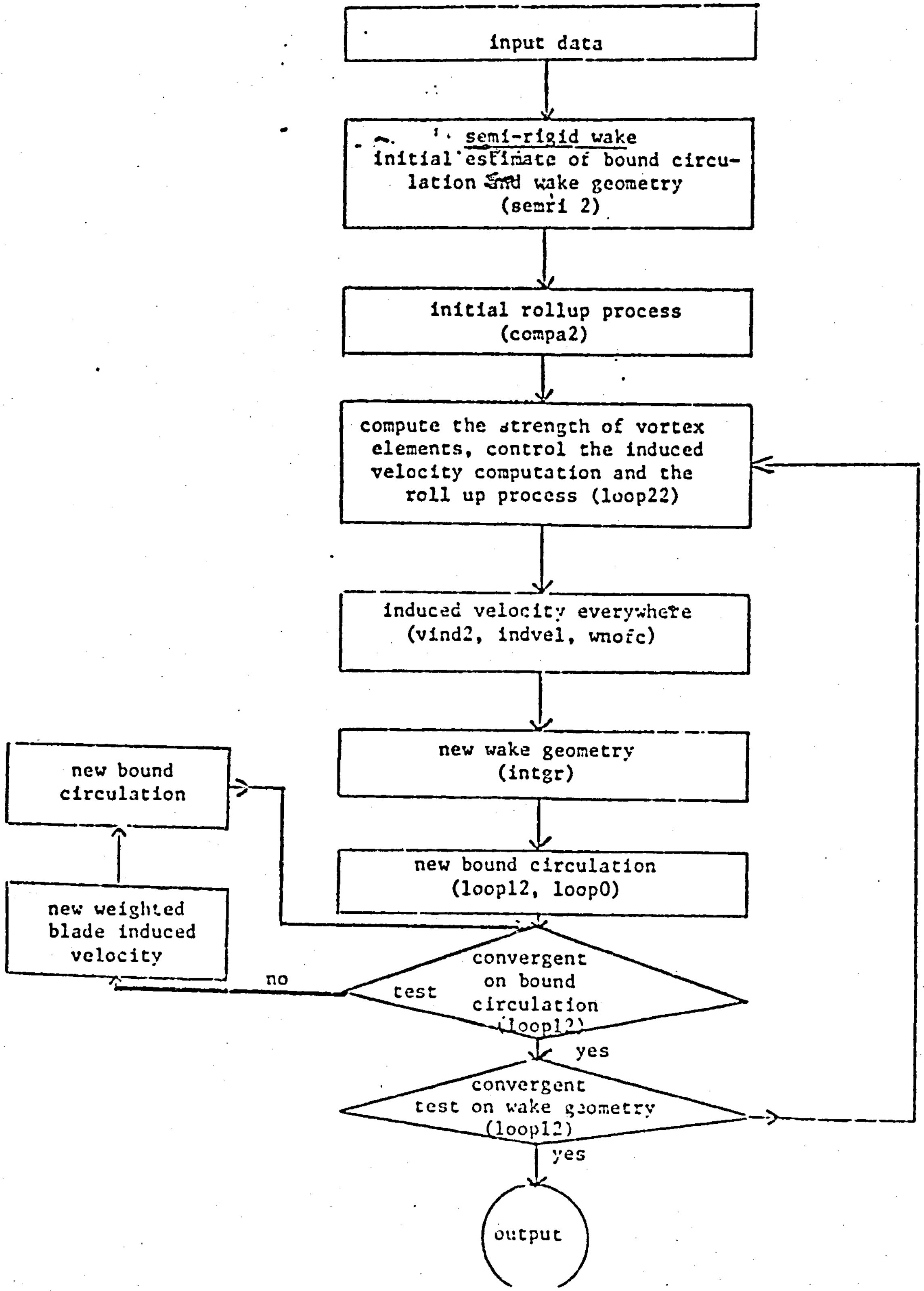


Table 6

```
$nparm
      lim2=20      $
$case
      nblds1=2, ltwist=1, knnvr=18, knivr=6,
      niva=35, sigma=0.0464, dpsind=10., dpsiid=20., alphas=.2, cd0=.014,
      cdk=.5, fps1=.03, fps2=0.01, coeff1=1.0, ntva=5, fmu=0.,
      thetad=16.95,15.3,14.2,13.65,12.55,10.9,10.46,9.91,9.69,9.25,8.7,8.4,
      8.15,7.93,7.71,7.49,7.22,7.05,
      etan=.1,.25,.35,.4,.5,.65,.69,.74,.76,.8,.85,.88,.9,.92,.94,.96,
      .985,1.0,
      etai=0.1,.25,.6,.85,.95,1.     $
```