General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
"Research on the Statically Thrusting Propeller"


NASA Grant No. NGL 39-009-172

Submitted by:

Joseph J. Eisenhuth
Associate Professor
Department of Aerospace Engineering

June 1978
Introduction

The impetus for this study came from the need to accurately predict the performance of propellers on V/STOL aircraft operating in the static condition. Small errors in thrust estimation are easily magnified into large errors in payload estimation. At the start of this study, it was felt that classical propeller analyses as well as some more recent numerical analyses methods did not adequately predict the static performance.

The classical vortex theory analyses for propellers are based on the physical situation of having the propeller advance at some finite forward velocity. In this theory each blade is modelled as a straight bound vortex filament and the wake behind each blade is represented by a force-free vortex sheet. For a lightly loaded optimum propeller, Betz\(^1\) showed that the geometry of each trailing vortex sheet is that of an uncontracted helical surface which is aligned with the resultant velocity in the slipstream. Goldstein\(^2\) was able to calculate the performance for the lightly loaded optimum propeller and Theodorsen\(^3\) later extended Goldstein's analysis to predict the performance of moderately loaded propellers, still retaining the true helical surfaces as the model of the trailing vortex sheets. When the classical methods are applied to the statically thrusting propeller, the predictions tend to be optimistic.

In the actual static propeller case, experimental evidence shows that the wake has a high degree of contraction, the vortex sheets near the tip tend to roll up into strong discreet vortices, and the inner part of each sheet tends to be distorted from the classical helical model.

In an attempt to more accurately model the static wake two other distinct approaches have generally been tried--the free-wake methods and
the prescribed-wake methods. The free-vortex approach is exemplified by Erickson and Ordway\(^4\). Their work is based on vortex theory in which they fix the wake contraction by means of heavily loaded actuator disk theory. The force-free condition for the trailing vortex sheets is then obtained by iterating on the pitch of these sheets. Characteristically this approach requires a large number of iterations and the final results are somewhat dependent upon the original chosen form of the vortex sheets.

The prescribed-wake analysis, is a semi-empirical approach exemplified by the works of Landgrebe\(^5\) and Ladden\(^6\). In this approach the wakes observed empirically are modelled and used to determine the induced velocity picture at the propeller blades. This kind of approach has the distinct advantage of using little computer time. These methods are, however, somewhat dependent upon the availability of experimental wake data.

In view of the need to eliminate any assumptions or empirical restrictions regarding wake shape, the main thrust of the present investigation has been to generate a wake without these restrictions and, thus, be able to calculate the induced flow at the propeller blades. To do this it is noted that the inflow is known exactly at one instant of time for any propeller; namely, at the instant of start of the propeller motion. Since no wake exists at this instant, the inflow is entirely determined by the blade motion. As the motion progresses, the wake is deposited and deforms continuously under its own self-induced effects until a final shape such as observed in Reference \(5\) is established. This means that the inflow and therefore the loading change continuously until the final wake is established and a steady state performance is reached. Essentially, the wake formation is treated as an initial condition problem in time. Such a formulation
implies an unsteady aerodynamic analysis for propellers similar to the Wagner problem of fixed-wing aerodynamics.

This report primarily summarizes the efforts toward this end. The principal recording of this work is in Reference 7. Prior to embarking on this study it was felt appropriate to develop a more standard performance calculation procedure to be used as a point of reference. The result was the somewhat modified prescribed wake procedure reported in Reference 8. Finally, in an effort to clear up some details of the program in Reference 7, particularly in regard to distortions of the vortex filaments that occur in the wake and the specification of core sizes for these filaments, Reference 9 was written.

A Reference Static Performance Method

In Reference 8, Miller reported on the development of a simple numerical method to rapidly predict the static performance of propellers. The wake model used in this development is essentially that of Gray\textsuperscript{10} who quantified the geometry of the tip vortex as a function of the performance of the propeller. However, Miller represented the rolled up tip vortices as a series of vortex rings whose position was consistent with the quantification proposed by Gray. In effect then the rings are used to calculate the axial component of induced velocity that would be produced by the rolled up tip vortices. The tangential component of induced velocity is derived by an analysis similar to that used in the classical vortex theory where the entire vortex sheet that is originally shed from each blade is taken into account. Finally, normality is imposed with regard to finding the additional
axial induced velocity that corresponds with the tangential component just calculated. This means that the vortex ring axial induced component is divorced from the normality consideration.

In order to evaluate the effectiveness of the developed method in predicting the performance of statically operating propellers, calculations were performed for several propellers for which static data were available. The first two configurations, Propellers I and II, were tested at Texas A & M University and the measured data were reported in Reference 11. The third configuration, Propeller III, was tested at the Wright-Patterson Air Force Base and the experimental results are reported in References 12 and 13.

Reasonably good correlation between the thrust and power predictions of the new method used and the measured data resulted in the case of Propellers I and II. Correlation for Propeller III was not so good as the other cases. However, according to Borst and Ladden\textsuperscript{14}, the Wright Patterson whirl rig that was used has a large cross-sectional area for the test rig relative to the area of the propeller disk. This kind of situation would certainly influence the wake geometry and could very well cause inaccuracies in the prediction method, if the wake geometry built into the method were used. Although an attempt was made to alter the description of the wake geometry by allowing for blockage, no great difference in the calculated results was realized and it was concluded that it would be necessary to generate empirical wake data for this particular case.

As a final part of the work with this prescribed-wake method, a sensitivity study was made relative to the various parameters used in defining the geometry of the ring stacks. It was found that the predicted performance is most sensitive to the axial spacing of the vortex rings. It was
found that the rate of contraction of the vortex rings had a relative effect in the thrust and power coefficient values but not in the Figure of Merit values. Although ways of improving the method were suggested in Reference 8, generally the computer program was found to be satisfactory for its intended purpose.

**Unsteady Vortex Lattice Technique Applied to the Wake Formation**

As previously noted, the principal record of the work with the unsteady vortex lattice technique was in Reference 7. In his work, Hall treated the blades as lifting surfaces, in fact, the treatment was general enough to handle either a propeller blade or a finite aspect ratio wing with only slight changes in the computer program.

The numerical model of the lifting surface and its wake consisted of replacing the continuous distribution of vorticity by a mesh of vortex segments of finite length and strength. The geometry of the wake vortices was fixed by the motion of an ever increasing number of points moving under the influence of the bound vorticity and its own self-induced effect since it was assumed that these wake points are connected by straight-line vortex segments identified as shed and trailing vorticity. The description of the blade bound vortices was fixed by the blade geometry.

The vortices on the surface were arranged in a conventional manner. The surface was broken into a number of spanwise segments and each spanwise segment was subdivided into a number of chordwise segments. Each resulting panel contained a control point and was spanned by a straight-line vortex segment. The spanning vortex was at the 1/4-segment chord of each panel and the control point was at mid-segment span and 3/4-segment chord of each...
panel. The final spanwise filament was 1/4-segment chord downstream of the surface trailing edge, implying that the Kutta condition was satisfied approximately, the accuracy of approximation increasing as the number of chordwise segments increase.

The surface loading was reflected in the strength of the bound vortices and the unique solution to the load distribution was determined by applying the boundary conditions of tangent flow at the surface and the Kutta condition. The solution was obtained numerically by expressing these conditions as a system of simultaneous algebraic equations and solving by matrix multiplication methods. The velocity associated with the vortices was described by the Biot-Savart law with the load distribution on the surface being determined from the unsteady Bernoulli equation.

At the start of this study considerable time was spent with the finite wing since this configuration contains essentially the same numerical problems as the propeller but is less complex. The first configuration that was tried was the infinite wing. This was done by making the aspect ratio sufficiently large (AR = 1000) that it adequately represented the infinite aspect ratio case. The comparison with the Wagner solution was quite good except in the initial instants where large deviations occur. The explanation for this lies in the fact that the Wagner solution contains only the effect of the wake whereas the numerical solution contains an "infinite" added mass lift solution with the impulsive start.

With regard to the finite wing, such things as the effect of the number of spanwise panels on the lift coefficient were investigated. Linearized wakes and wakes which distorted under the influence of velocities induced by the shed, trailing, and bound vortex systems were also studied. It was
found generally that, for the distorted wake case, there was little tendency
for the vortices to roll up into tip vortices unless localized induction
was taken into account. The extreme slowness of the roll up rates was
remedied by the use of localized induction concept\textsuperscript{15} which, in effect, says
that a curved vortex filament induces at a point on itself a velocity propor-
tional to the local curvature and is directed along the local binormal.
This means that the curvature of the trailing vorticity can induce a span-
wise flow which will tend to destroy the initial two-dimensional character
of the motion. Under this influence the vortex segment end points describing
the wake will travel spiral paths which promote interference between
filaments and increase the roll up rate. It was concluded that this
localized induction effect is an essential ingredient as far as a realistic
roll up process is concerned.

The Biot-Savart equation contains a singularity, if the point at which
the induced velocity is determined lies on the filament. In order to circum-
vent this, the common practice is to assume a core exists in which the fluid
moves as a solid body and not as potential flow. Studies were made by Hall
with regard to the proper core size to use. Along with this kind of study
Hall also assumed that the circulation about a vortex segment varied with
the length of the segment as it distorted. It is with the idea of core
size and the circulation as the segments distort that Dasso\textsuperscript{9} was primarily
interested.

With the analysis established and verified for the finite wing, Hall
undertook the case of the statically thrusting propeller. He tried
predictions of a four-bladed propeller whose performance was presented in
Reference 16. It was found that the theoretical results for the propeller
configuration did not correlate well with the experimental results. In an
In an effort to obtain further comparisons, a classical Prandtl analysis was performed and calculations based on momentum theory were made. In general, reasonable comparisons in thrust predictions were obtained between Hall's analysis and the Prandtl analysis while the actual measurements of Reference 16 were considerably lower. Then, using an average thrust coefficient, $C_T$, of the value predicted by either the present analysis or the Prandtl analysis, a momentum power coefficient, $C_{pi}$, was calculated. It was found that Hall's analysis compared favorably with this $C_{pi}$ value as well as with the total $C_p$ of the Prandtl analysis. However, all of these calculated values are much higher than the measured $C_p$ of Reference 15, indicating that possibly the measured $C_p$ was too low. The error observed in these results was much greater than anticipated, particularly since the finite wing results were so encouraging.

Further error in the analysis could have been due to poor synthesis of the airfoil section data. Although care was taken and the guidelines of Reference 16 were followed, the airfoil section was nonstandard and difficult to describe. Poor estimates of the drag characteristics could explain, in part, discrepancies in the power calculations among analyses with reasonable thrust comparisons.

Error might also have been due to the relatively short wakes generated. Even though extremely long computational run times (20,000 sec.) were performed, only about two revolutions of wake could be generated at best, and it is quite conceivable that this is not enough to predict the steady state performance. It was noted that the average $C_T$ and $C_{pi}$ responded to the impulsive start much like a low aspect ratio wing. That is, following the impulsive start the performance dropped very quickly to what appears to
be the steady state value. It is possible that steady state had not been attained and more revolutions were necessary. This would lead to an increased inflow which, by decreasing blade angle of attack, could lead to decreased thrust prediction. Regions of inboard stall would be determined by this inflow, and performance would be measurably affected by the extent of these regions.

Finally, there is an error due to the vortex wakes deposited by the propeller blades. The time steps considered were generally much too large to predict accurate wakes. As a result, the wakes of the four-bladed configuration became unstable; this instability was enhanced by interaction core radii that were too small. The resulting wake geometries then contained extremely long straight line vortex segments which, once formed by a strong interaction induced velocity acting over a relatively large time step, could produce completely erroneous velocities at the blades. To make matters worse these segments could never return to a reasonable geometry as time progressed since they might never pass through enough interactions to counteract the effect of one strong one. It should be noted that wake instabilities noted in the analysis are believed to be only numerical with no physical counterpart.

Even though the comparison of theoretical and experimental results leaves much to be desired, some parametric results were successfully obtained. Small time steps (1.5° to 3° in azimuth) are required for accurate wake prediction. This is necessary to determine an accurate vortex filament radius of curvature for calculating the locally induced effects. This is also a requirement in order to obtain reasonable vortex induced curved paths for
the wake points from the one-step Euler integration scheme which can only provide straight line translation of a point.

One of the important regions of concern is the location where the wake from a preceding blade comes close to the following blade. The accuracy of these blade-wake interactions not only depends on small time steps, but also on interaction core radii large enough to limit the movement of a wake point to a reasonable value.

The conclusion of this work must admit that the accuracy of the present analysis when applied to the statically thrusting propeller has not been satisfactorily demonstrated since correlation with the selected experimental results was poor. Even though the basic formulation is believed sound from comparison with other analyses and finite wing results, final correlation will have to await better experimental results, more accurate airfoil section characteristics, relief from the numerical inaccuracies associated with the aerodynamic interference region and larger computational runs to numerically establish the wake. This procedure, like other vortex lattice techniques, uses an inordinate amount of computer time due to the repeated calculations of the Biot-Savart Law in the wake. Unfortunately, no wake simplification or approximations are apparent because of the importance of the nonlinear flow of the induced velocity field. This is further aggravated by the small time step requirement to compute interference aerodynamics of the problem accurately. This seriously restricts the usefulness of the analysis, at present even as a research tool. However, vortex lattice techniques are those which most readily apply to nonlinear aerodynamic problems so that further attempts at reducing the computation time of this analysis, as
well as accepting long time computer runs, are perhaps justified, at least in research problems.

In spite of the inconclusiveness of the primary results of this analysis, some positive results were obtained. Perhaps the most significant of these is the modeling of the wake roll-up with the localized induction concept while considering the three-dimensional flow about a lifting surface starting from rest.

Vortex Core Size Study

As mentioned earlier, the vortex core size and how the circulation varies with elongation of the vortex segments was of concern in this general study. Daso was concerned with providing some rational approach to determining core sizes which wasn't just an arbitrary choice. He was also concerned with keeping track of core sizes as the segments distorted and this aspect of the problem was intimately related to what happens with the circulation during distortion.

In general he showed that the circulation must remain constant regardless of segment length. On the other hand, because of the conservation of mass in the core, the vorticity will increase with increasing length of the vortex segment and the core radius will correspondingly decrease. Therefore, once having established a core size, it is a matter of routine to keep track of the core size as the vortex segments distort.

With regard to establishing the initial core radius, Daso first looked into an approach which took cognizance of the fact that at the trailing vortex sheet the velocity induced just above and just below the sheet are proportional to the circulation per unit length. By dividing the sheet...
into sections, the circulation per unit length is known from the circula-
tion distribution and in turn the induced velocity is known. The vortex
core size can then be found via the Biot-Savart law, but since the induced
velocity depends on the original arbitrary chord of sheet segment size,
the vortex core that results is also arbitrary.

Daso then studied the pressure distribution for a Rankine vortex with
the thought that, by using reasonable values of minimum pressure at the
center of vortex, the core radius could be explicitly calculated. Although
the minimum pressure at the center of the core is generally unknown and
cannot be theoretically determined, plots of core radius versus minimum
pressure show a range of pressure where little change in core radius occurs.
Beyond this range the core radius increases quite rapidly and a zero
minimum pressure in the other direction is an improbability. On this basis
an average value of minimum pressure of about 400 lbs per square foot was
chosen. The error in radius involved in this choice varies from about
8 percent at a minimum pressure of 100 lbs per square foot to 15 percent
at 800 lbs per square foot. Although it was recognized that there was
a degree of arbitrariness in the choice of core size, it was reasoned that
this degree was relatively small. With the minimum pressure chosen, the
core size becomes a function of circulation. Daso made studies of how the
initial estimate of core radius varied with forward velocity, spanwise
position, and with time step. He did this for both the trailing and shed
vortex segments. He did, however, restrict himself to the use of the
program with wings only because, as in Hall's work, it is much easier to
work with wings rather than the propeller when proofing such features as
just discussed.
The question of how the changes made in the program would affect the final results when running the propeller case is still unanswered. Time and money did not permit such an exercise. A listing of the general computer program is presented in the Appendix. The code contains comment statements for parts of the program that were used in the vortex core study just described. These can be included in the program by just eliminating the comment designation. The same can be said for a number of other statements which give the option of running program with the IBM 370 facilities at Penn State or at the NASA Langley facilities.

Conclusions

In spite of the generally favorable trends established from applying vortex lattice techniques to the statically thrusting propeller, the primary objective of obtaining the high degree of accuracy necessary to correlate theory and experiment has not been accomplished. However, the major problem areas in the aerodynamic modeling have been identified and the foregoing analysis represents a tool to investigate these areas.

If more fruitful results are to be obtained, efforts to reduce computer time must be of prime concern. Attempts to more accurately predict the potential inflow lead to small time increments corresponding to an azimuth step size, $\Delta \theta < 1.5^\circ$, fully one-half the smallest value considered and at least one tenth a value at present practical. This limit has been established by estimates necessary to promote good wake roll-up characteristics. Attempts in the present analysis to reduce computer central processor time (and core storage) with special data handling techniques have been generally unfruitful.
Reductions in computation time would also permit more accurate representations of the wake. The numerical integration scheme considered in the present analysis is a simple one-step Euler scheme, shown to be less accurate than either a Runge-Kutta method or a one-step predictor-corrector technique. The inherent inaccuracy of the method lies in the fact that points can only translate under the influence of a vortex induced velocity whereas the true path is circular. Unfortunately this method is the most economical from the point of view of computation time and core storage, although to get a sufficiently close approximation to the circular path requires very small time increments.

The final work on core size and the variation of vorticity in the core appears to have resulted in a satisfactory means of handling these quantities.
REFERENCES


APPENDIX

Computer Program for the Unsteady Vortex Lattice Technique

The following is a listing of the computer code used in the unsteady vortex lattice approach. As can be seen in the following description of input data, certain choices permit the program to be run for a wing or a propeller. For instance, the choice of zero rpm and one blade permits the analysis of a wing.

First Data Card

NOPAN - Number of spanwise panels of the lifting surface.
NUM - Number of spanning vortices including the shed vortex on each spanwise panel.
MXTIME - Maximum number of time steps.
IBL - Number of blades.

Second Data Card

V - Forward or free-stream velocity.
RPM - Revolutions per minute.
R - Radius of blade or wing span.
BL - Number of blades.
DELT - Angular increment of blade travel in one time step.

Third Data Cards

YC(L) - Control point coordinate along spanwise or Y-axis of Lth control point.
BETAC(L) - Pitch angle at a control point or angle of attack of wing at the Lth control point.
THC(L) - Section thickness at a control point.

CC(L) - Chord length of a lifting surface element corresponding to Lth control point.

XRC(L) - X location with respect to the leading edge of the stackup point or origin of the blade based coordinate system.

Fourth Data Cards

The input variables, Y(L), BETAL(L), TH(L), C(L), and XR(L) have similar definitions to those of the third data cards except that they refer to the edge of the lifting surface panel containing the control point, L.

Other Input Variables

RHO - Nondimensional fluid density.

ROH - Density of air.

A sample of the input data used in the case of a wing study appears on the last page, following the program listing. The occasional printing of CANADAIR in the program is in reference to propeller section data for a CANADAIR propeller.
C

PROGRAM PENNST(INPUT, OUTPUT, TAPE5=INPUT, TAPE1)
DOUBLE PRECISION DARG

C
REAL ITIMUZ
INTEGER STATUS
DIMENSION Q(100, 1), IPIVOT(100), INDEX(100, 2)
DIMENSION A(100,100), F(100)
DIMENSION X(10,30), Y(30), Z(10,30), VX(120), VY(120), VZ(120)
DIMENSION Z(10,30), WXC(1C,30), WYC(10,30), WZC(10,30), FXQS(10,30),
FYQS(10,30), FZQS(10,30), FXUS(10,30), FYUS(10,30), FZUS(10,30),
1 THRST(30), DRAG(30), TORQ(30)
DIMENSION XR(30), XRC(30), DZP(20,30), TH(30), THC(30)
DIMENSION V(30,50), VJ(30,50), VK(30,50)
DIMENSION FXBV(10,30), FYBV(10,30), FZBV(10,30)
DIMENSION PQS(10,30), PUS(10,30), P(10,30), POWR(30)
C
DIMENSION PHN(30), SUM(30)
DIMENSION D(3), STATUS(4)
DIMENSION SNINAM(10,30), COSLAM(10,30)
COMMON X(10,30), Y(30), Z(10,30), VX(30,100), VY(30,100), VZ(30,1)
1, GAMMA(100), GAM(30,100), GAM(30,1), GAM(30), AN(30), RA(30), QA(30), C(3)
1, BETA(30), AS(30), UT(30,31), YVC(30), CC(30), BEAC(30), RAS(30,100),
1, RAT(30,100)
COMMON XNP, YNP, ZNP, COT, SIT, ITIME, IWAKE, XBL, YBL, NBP, NUM, XD, YD, ZD,
1, E, AD, R, KMAX, LINWA, V, SUMARR
DATA STATUS/40/
1000 FORMAT(1H,9(E13.5))
1001 FORMAT(1H, S) THRUST DISTRIBUTION', 10X, 'POWER DISTRIBUTION', 7X,' 1'EFFECTIVE ALF.A DISTRIBUTION'
1002 FORMAT(1H, S) THRUST COEFFICIENT', 10X, 'POWER COEFFICIENT', 10X,
1'PROPELLER CONVENTION'
1003 FORMAT(' ',',',',E13.5)
1004 FORMAT(1H0)
1005 FORMAT(', I5,6(E13.5), I5)
1006 FORMAT('O',9(E13.5, 1X))
1007 FORMAT('O',2(I5,2X))
1008 FORMAT(' ', 10X, 'TIME USED=', I10)
1009 FORMAT(' ', 5(10X,E15.8))
1010 FORMAT('O',10X,2(E13.5,5X))
1011 FORMAT('O',10X,1E15.8)
1012 FORMAT(7(15))
1013 FORMAT(7F10.6)
1014 FORMAT(' ', 'MAXIMUM NUMBER OF TIME STEPS IS ', I5)
1016 FORMAT('O',12('CANADAIR '))
1017 FORMAT('O', 'NUMBER OF SPANWISE PANELS=', I3//' NUMBER OF CHORDWISEPROG065
1E VORTICES=', I3//' MAXIMUM NUMBER OF TIME STEPS=', I4//' PROG070
2 BLADE NUMBER=', I2)
1018 FORMAT('O', 'FLIGHT SPEED=', E12.5, 'FEET PER SECOND RPM=', PROG075
1E12.5//' RADIUS(SPAN)=', E12.5, ' NUMBER OF BLADES=', E12.5//' PROG080
2 DELTA THETA=', E16.8,'DEGREES')
1019 FORMAT('O', 'TIME INCREMENT=', E16.8//' REL. LENGTH OF CHORDWISE PROG090
1 PANEL=', E12.5//' COEFFICIENT=', E16.8)
1020 FORMAT('O', 'BLADE SECTION CHARACTERISTICS 'PROG0500
1// CONTORL POINT GEOMETRY 'PROG0505
2// YC/R 'PROG0506

ORIGINAL PAGE IS
OF POOR QUALITY.
A-G

1021 FORMAT(' ',I6,E13.5,9X)
1022 FORMAT(' ',VORTEX GEOMETRY
1//', Y/R, BETA(DEG.), T/C
2C/R, X/R')
1023 FORMAT('O', BOUND VORTICITY DISTRIBUTION'/)
1050 FORMAT(' ',*** PHI=',E16.8)
7777 FORMAT(1H, 'SHED VORTEX CORE RADIUS, RAS', 5X, 'TRAILING VORTEX COR
1E RADIUS, RAT')
5555 FORMAT('O', 'MINIMUM PRESSURE, PMIN LBS PER FT**2')
6666 FORMAT('O', 20X, 2(E13.5,2OX))
8888 FORMAT(1H, 3(E13.5,20X))

C LINEARIZED BOUNDARY CONDITION REMOVED
C LINEARIZED OR DEFORMED WAKE
C LINWA=1 IMPLIES LINEARIZED WAKE
C LINWA=ANYTHING ELSE IMPLIES DEFORMED WAKE
C
C NOTE** IN FREE WAKE ANALYSIS, CONSERVATION OF CIRCULATION HAS
C GAMT*AL=CONST AT SHEDDING (FUNC. ONLY OF TIME OF SHEDDING AND
C SPANWISE POSITION). 2 ( B-S LAW GAM=(GAMT*AL)/AL(T)=CONST/AL
C WHICH LEADS TO AL**2=ALS APPEARING IN DENOMINATOR INSTEAD OF AL
C
C CALL TIMUSE(ITIMUZ)
PRINT1008,ITIMUZ
C CALL BLKREWD(5LTAPI)
PRINT 1016
LINWA=0
C LINWA=1
READ(5,1012) NOPAN, NUM, MXTIME, IBL
PRINT 1017, NOPAN, NUM, MXTIME, IBL
READ(5,1013) V, RPM, R, BL, DELTH
PRINT 1018, V, RPM, R, BL, DELTH
RHO=1.
BL=IBL
PI=3.1415927
C**** MDIM MUST BE GREATER THAN OR EQUAL TO FIRST SUBSCRIPT OF ARRAY A
C*** (MAIN PROGRAM), SO THAT ARRAY A (SUBROUTINE MXINV) IS PROPER
MDIM=100
CED
ITIME=0
HH=0.001
E=0.000001
H=HH
C**** V=0 IMPLIES HOVER
C*** THETA=0 IMPLIES PSI=90
THETA=0.
RPS=RPM*PI/30.
VTIP=RPS*R
DELT=DELT/PI/180.
COEF=RHO*PI*R*R*VTIP*VTIP
IF(COEF.NE.0.0) GO TO 107
C COEF=0.5*RHO*V*V*R*R/3.0
C
DELT=DELT/RPS
107 DELT=0.1
C**********************************************************NOPAN=NUMBER OF SPANWISE PANELS
C**********************************************************NUM=NUMBER OF CHORDWISE VORTICES, INCLUDING SHED AT T.E.
C**********************************************************MXTIME=MAXIMUM NUMBER OF TIME STEPS IN WAKE
C**********************************************************V=FLIGHT SPEED
C********************************************************** RPM=RPM
C********************************************************** R=RADIUS, SPAN
C********************************************************** BL, IBL=NUMBER OF BLADES
C********************************************************** DELTH=ANGULAR INCREMENT OF BLADE TRAVEL IN ONE TIME STEP
C********************************************************** RHO=FLUID DENSITY
C********************************************************** COEF=NONDIMENSIONALIZATION FACTOR FOR FORCES
C********************************************************** DELT=TIME STEP INCREMENT

NUMM1=NUM-1
NPANP1=NOPAN+1
NPANM1=NOPAN-1
NPANP4 = NOPAN + 2
MATRX1=NUM*NOPAN
MATRX3=NUMM1*NOPAN
MATRX2=MATRX3+1
AO=0.1076
DELX=1./NUMM1
PRINT 1019, DELT, DELX, COEF
PRINT 1020
DO 113 L=1,NOPAN
  READ(5,1013)YC(L),BETAC(L),THC(L),CC(L),XRC(L)
  PRINT 1021,YC(L),BETAC(L),THC(L),CC(L),XRC(L)
  BETAC(L)=BETAC(L)*PI/180.
  YC(L)=YC(L)*R
  CC(L)=CC(L)*R
113 CONTINUE
PRINT 1022
DO 114 L=1,NPANP1
  READ(5,1013)Y(L),BET(A(L),TH(L),C(L),XR(L)
  PRINT 1021,Y(L),BET(L),TH(L),C(L),XR(L)
  BET(L)=BET(L)*PI/180.
  Y(L)=Y(L)*R
  C(L)=C(L)*R
114 C(L)=C(L)*R
DO 115 L=1,NOPAN
  DO 115 I=1,NUMM1
    DZP(I,L)=0.
100 DO 116 L=1,NOPAN
    DO 116 I=1,NUMM1
      WX(I,L)=0.
    WY(I,L)=0.
116 WX(I,L)=0.
117 WX(I,L)=0.
C*** XC IMPLIES CONTROL POINT, VORTEX COORDINATE WRT BLADE SYSTEM
DO 3 L=1,NOPAN
  DO 3 I=1,NUMM1
    RI=I
    XC(I,L)=((RI-.25)*DELX-XRC(L))*CC(L)*COS(BETAC(L))
    ZC(I,L)=-(RI-.25)*DELX-XRC(L))*CC(L)*SIN(BETAC(L))
DO 2 L=1,NPANP1
DO 2 I=1,NUM
RI=I
X(I,L)=((RI-.75)*DELX-XR(L))*C(L)*COS(BETA(L))
2 Z(I,L)=-(RI-.75)*DELX-XR(L))*C(L)*SIN(BETA(L))
C**DETERMINATION OF LOCAL DIHEDRAL****
DO 600 L =1,IVOPAN
DO 600 I=1,NUMM1
ALAM=-ATANI-XC(I,L)*((TAN(BETA(L+1))-TAN(BETA(L)))/(Y(L+1)-Y(L)))
600 COSLAM(I,L)=COS(ALAM)
21 II=0
C** COEFFICIENTS DETERMINED ON BASIS OF STRIP THEORY-NO SPANWISE EFFECTS
DO 4 L=1,NOPAN
CCOSBL=COS(BETA(L))*CC(L)
CSINBL=CSIN(BETA(L))*CC(L)
DO 4 I=1,NUMM1
XD=XCI
YD=YCI
ZD=ZCI
C** ZN,XN,YN UNIT NORMAL COMPONENTS WRT BLADE FIXED AXIS
ZN=CCOSBL*COSLAM(I,L)-DZP(I,L)*COSLAM(I,L)
XN=CSINBL*COSLAM(I,L)
YN=CSINLAM(I,L)
II=II+1
JJ=0
DO 4 K=1,NOPAN
DO J=1,NUM
JJ=JJ+1
A1=0.
A2=0.
A3=0.
DO 11B=1,IBL
CSIN=COS(2.*PI*(IB-1)/BL)
SSIN=SIN(2.*PI*(IB-1)/BL)
DO 1 KKK=1,3
IF(J-NUM) 111,112,111
111 IF(KKK-2) 109,112,110
109 XB=X(J,K)*CSIN-Y(K)*SSIN
XA=X(NUM,K)*CSIN-Y(K)*SSIN
YB=Y(K)*CSIN+X(J,K)*SSIN
YA=Y(K)*CSIN+X(NUM,K)*SSIN
ZB=Z(J,K)
ZA=Z(NUM,K)
GO TO 106
112 XB=X(J,K+1)*CSIN-Y(K+1)*SSIN
XA=X(NUM,K)*CSIN-Y(K+1)*SSIN
YB=Y(K+1)*CSIN+X(J,K+1)*SSIN
YA=Y(K)*CSIN+X(J,K)*SSIN
ZB = Z(J, K+1)
ZA = Z(J, K)
GO TO 108

110 XB = X(NUM, K+1) * CSIN - Y(K+1) * SSIN
XA = X(J, K+1) * CSIN - Y(K+1) * SSIN
YB = Y(K+1) * CSIN + X(NUM, K+1) * SSIN
YA = Y(K+1) * CSIN + X(J, K+1) * SSIN
ZB = Z(NUM, K+1)
ZA = Z(J, K+1)

108 CONTINUE
XBA = XB - XA
YBA = YB - YA
ZBA = ZB - ZA
XDA = XD - XA
YDA = YD - YA
ZDA = ZD - ZA
XDB = XD - XB
YDB = YD - YB
ZDB = ZD - ZB
ALS = XBA * XBA + YBA * YBA + ZBA * ZBA
ACS = XDA * XDA + YDA * YDA + ZDA * ZDA
BCS = XDB * XDB + YDB * YDB + ZDB * ZDB
AL = SQRT(ALS)
AC = SQRT(ACS)
BC = SQRT(BCS)
COSA = (ACS + ALS - BCS) / (AC * AL * 2.)
COSB = (BCS + ALS - ACS) / (BC * AL * 2.)
TEMPA = 1. - COSA * COSA

117 VFN = (COSA + COSB) / (AL * ACS * TEMPA * PI * 4.)
AILXAC = YBA * ZDA - ZBA * YDA
AJLXAC = ZBA * XDA - XBA * ZDA
AKLXAC = XBA * YDA - YBA * XDA
A1 = A1 + VFN * AILXAC
A2 = A2 + VFN * AJLXAC
A3 = A3 + VFN * AKLXAC
IF(J - NUM) 1, 4, 1
1 CONTINUE
4 A1(J1, JJ) = A1 * XN + A2 * YN + A3 * ZN
GO TO 1193

1193 PO = 2116.8
ROH = 0.002378
C PMIN = 600.0
C DO 1191 J1 = 1, 10
C PMIN(J1) = J1 * 100.0 - 100.0
C SUMAR(J1) = SQRT((PO - PMIN(J1)) / ROH)
C SUMARR = SUMAR(J1)
C PRINT 6666, PMIN(J1)
C DO 88 L = 1, NPANP1
C NPANP3 = L
C SUMD = 0.0
C DO 87 N = 1, NPANP3
C BA = 2.0 * N - 1.0
A-B

C  SN=SORT(BA)
C 87 SUMD=SUMD+AN(N)*SN
C  RAIL(L) = ( V*R *SUMD)/SUMA(J)
C PRINT 8888, AN(L), RAI(L)
C 88 CONTINUE

K=1
D011 I=1, MATRX2, MATRX1
LLL=K+SUM
D012 JJ=1, MATRX1
A(I, J)=0.
IF(JJ.GE.KK AND JJ.LT.LLL) A(I, JJ)=1.
12 CONTINUE
11 K=K+SUM
C DO 100 I=1, MATRX1
C 100 PRINT 1000, (A(I, J), J=1, MATRX1)
C CALL MATINV(A, MATRX1, Q, O, DETERM, IPIVOT, INDEX, 100, ISCALE)
CALL MXINV(A, MDIM, MATRX1)
C PRINT 1011, DETERM
C PR NT 1012, ISCALE
CE PMIN=0.0
PMIN=400.0
PRINT 5555
PRINT 6666, PMIN
C3003 CONTINUE

ITIME=0
SUMARR=SQRT((PO-PMIN)/ROH)
71 CONTINUE
CALL TIMUSE(ITIMUZ)
PRINT1008, ITIMUZ
RTIME=ITIME
TIME=RTIME*DELTH
THETA=DELT*RTIME
SINTH=SIN(THETA)
COSTH=COS(THETA)
C******** IMPACT VELOCITY--ITIME=0 **********
C*** COMPUTATION OF IMPACT VELOCITY, VN(I) ***
II=0
DO 25 L=1, NOPAN
DO 25 I=1, NUMM1
II=II+1
25 VN(I)=(RPS*YC(L)+V*COSTH)*SIN(BETAC(L)-DZP(I, L))*COSLAM(I, L)+(-RP
15*XG(L, L)-V*SINTH)*SINLAM(I, L)
IF(ITIME)14, 19, 20
19 DO6 II=I, MATRX3
6 F(I):=-VN(I)
DO 27 II=MATRX2, MATRX1
27 F(I)=0.
GO TO 29
20 CONTINUE
C*** WAKE BOUNDARY VELOCITIES WRT BLADE-FIXED SYSTEM ***************
II=0.
CED PRINT 7777
DO8 L=1, NOPA)
**BACK TRANSFORM WAKE COORDINATES TO BLADE -FIXED SYSTEM************

\[
\cot = \cos \theta \\
\sin = -\sin \theta \\
I_{\text{wake}} = 1 \\
H = H \\
\text{CALL INVEL}
\]

IF (KMAX .EQ. NUM) GO TO 1149
IF (KMAX .EQ. NPANPI) GO TO 1149
IF (L .GT. 1) GO TO 1149
IF (II .GT. I) GO TO 1149

1132 PRINT 8888, AN(K1), RA(K1)
1132 PRINT 8888, 014AS(K1), RAT(K1, ITIME + 1)
1149 WXC(I, L) = VXP
      WYC(I, L) = VYP
      WZC(I, L) = VZP

**BOUNDARY CONDITION FROM STRIP THEORY**

\[
F(I) = -V_{N}(I) - V_{ZP} \cos(BETAC(L) - DZP(I, L)) \cos \lambda(I, L) - V_{XP} \sin(BETAC(L) - DZP(I, L)) \sin \lambda(I, L)
\]

**KUTTA CONDITION**********

II = MATRX3
DO 16 L = 1, NOPAN
      II = II + 1
      M = (L - 1) * NUM + 1
      N = M + NUM - 2
      SUM = 0.
      DO 17 JJ = M, N
      SUM = SUM + GAMMA(JJ)
      17 F(I) = SUM
      16 CONTINUE

C  PRINT 1004
C  PRINT1000, (F(I), I = 1, MATRX1)
C  CALL MXMLT(A, F, GAMMA, MATRX1, MATRX1, 1, 100, 100, 100)

**BOUND VORTICITY OUTPUT**********

PRINT 1023
DO 18 L = 1, NOPAN
      M = (L - 1) * NUM + 1
      N = M + NUM - 1
      18 PRINT 1000, (GAMMA(II), II = M, N)
PRINT 1004

**FORCE DUE TO DELTA-P—CHORDWISE LOADING

**FORCE DUE TO DELTA-P—CHORDWISE LOADING

**FORCE ON PANEL DUE TO DELTA-P

DO 33 L = 1, NOPAN
      M = (L - 1) * NUM + 1
      N = M + NUM - 2
      I = 0
C** M,N LOCATE CONTROL POINTS AND PANELS
C*** L,I=CONTROL POINT INDICES
DO 34 I=M,N
   I=I+1
C*** DETERMINATION OF QUASI-STATIC FORCE ON PANEL(I,L)
C*** GAM1=CHORDWISE VORTICITY ON LEFT OF CONTROL POINT
C*** GAM3=CHORDWISE VORTICITY TO RIGHT OF CONTROL POINT
C*** GAM1,GAM3(+) FEEDING INTO TRAILING EDGE
   GAM1=0.
   GAM3=0.
C
   GO TO 601
   DO 137 J=M,II
      IF(L-1) 138,138
   134 GAM1=GAM1+GAMMA(J)
      GAM3=GAM3+GAMMA(J+NUM)-GAMMA(J)
      GO TO 137
   138 IF(L-NOPAN) 141,140,141
   140 GAM1=GAM1+GAMMA(J)-GAMMA(J-NUM)
      GAM3=GAM3-GAMMA(J)
      GO TO 137
   141 GAM1=GAM1+GAMMA(J)-GAMMA(J-NUM)
      GAM3=GAM3+GAMMA(J+NUM)-GAMMA(J)
   137 CONTINUE
   (31 CONTINUE
      GAM2=GAMMA(II)
C*** GAM2(+) LEFT TO RIGHT
C ******************* INITIALIZATION OF FORCES *******************
   FXQS(I,L)=0.
   FYQS(I,L)=0.
   FZQS(I,L)=0.
   PQS(I,L)=0.
   DO 142 KKK=1,3
      IF(KKK=2) 144,145,146
   144 XD=(X(I,L)+X(I+1,L))/2.
      YD=Y(L)
      ZD=(Z(I,L)+Z(I+1,L))/2.
      J=I
      K=L
      JJ=I+1
      KK=L
      GAMA=GAM1
      GO TO 148
   145 XD=(X(I,L)+X(I,L+1))/2.
      YD=(Y(L)+Y(L+1))/2.
      ZD=(Z(I,L)+Z(I,L+1))/2.
      J=I
      K=L+1
      JJ=I
      KK=I
      GAMA=GAM2
      GO TO 148
   146 XD=(X(I,L+1)+X(I+1,L+1))/2.
YD = Y(L+1)
ZD = (Z(I,L+1) + Z(I+1,L+1))/2.
J = I
K = L+1
JJ = I+1
KK = L+1
GAMA = GAM3

148 IAWAKE = 0
VX = RPS*YD + V*COSTH
VY = -RPS*XD - V*SINTH
VZ = 0.

C************ VELOCITIES DUE TO BOUND VORTICITIES CALCULATED IN BLADE -FIXED SYSTEM
C

151 CONTINUE
CALL INVFL
C
IF (KMAX.EQ.NUM) GO TO 1150
C
C IF (KMAX.EQ.NPANPI) GO TO 1150
C
C IF (L.GT.1) GO TO 1150
C
C IF (II.GT.I) GO TO 1150
C
C IF (KKK.GT.1) GO TO 1150
C
C PRINT 7777
C DO 1133 K2 = L, NPANPI
C
1133 PRINT 8888, AN(K2), RA(K2)
C
1133 PRINT 8888, RAS(K2, ITIME+1), RAT(K2, ITIME+1)

1150 VX = VX + VXP
VY = VY + VYP
VZ = VZ + VZP

VPOWX = 2. * (RPS*YD + V*COSTH) - VX
VPOWY = 2. * (-RPS*XD - V*SINTH) - VY
VPOWZ = -VZ

IF (ITIME.EQ.0) GO TO 150
IF (IWAKE.EQ.1) GO TO 150, 150

149 IAWAKE = 1

C***** BACK-TRANSFORM WAKE COORDINATES TO BLADE -FIXED SYSTEM ***********************
C
C
GOTO 151

150 CONTINUE
FXQS(I,L) = RHC*(VY*(Z(J,K) - Z(JJ,KK)) - VZ*(Y(K) - Y(KK))) * GAMA
1+FXQS(I,L)
FYQS(I,L) = RHC*(VZ*(X(J,K) - X(JJ,KK)) - VX*(Z(J,K) - Z(JJ,KK))) * GAMA
1+FYQS(I,L)
FZQS(I,L) = RHC*(VX*(Y(K) - Y(KK)) - VY*(X(J,K) - X(JJ,KK))) * GAMA
1+FZQS(I,L)

PQS(I,L) = FXQS(I,L)*VPOWX + FYQS(I,L)*VPOWY + FZQS(I,L)*VPOWZ
1+PQS(I,L)
C
PRINT9999, I, L, KKK, XU, YD, ZD, VY, VZ, GAMA
9999 FORMAT(*, '1,3(1X, 13),7(1X, E15.8))
CONTINUE

**PRECEDING QUASI-STATIC FORCES HAVE LEADING EDGE SUCTION**

**DETERMINATION OF UNSTEADY FORCE ON PANEL(I,L)**

SUMP = 0.0
DO 35 J = M, II
  IF(I(1) TIME) 36, 36, 37
36 SUMP = SUMP + GAMMA(J)
  GO TO 35
37 SUMP = SUMP + GAMMA(J) - TGAM(J)
35 CONTINUE
SUMP = SUMP / DELT * RHO

**UNIT NORMAL COMPONENTS AT XC(I,L) FROM (AL)*X(AC)/((AL*AC) WITH**

**AL = SPANWISE VORTEX SEGMENT AND AC = (L+1) CHORDWISE SEGMENT**

XBA = X(I,L+1) - X(I,L)
YBA = Y(L+1) - Y(L)
ZBA = Z(L+1) - Z(I,L)
XDA = X(I,L+1) - X(L,L+1)
ZDA = Z(I,L+1) - Z(L+1,L+1)

AL = SQRT(XBA * XBA + YBA * YBA + ZBA * ZBA)
DARG = XBA * XBA + YBA * YBA + ZBA * ZBA
AL = DSQRT(DARG)
AC = SQRT(XDA * XDA + ZDA * ZDA)
DARG = XDA * XDA + ZDA * ZDA
AC = DSQRT(DARG)

AILXAC = YBA * ZDA
AJLXAC = ZBA * XDA - XBA * ZDA
AKLXAC = YBA * XDA
DARG = AILXAC * AILXAC + AJLXAC * AJLXAC + AKLXAC * AKLXAC
ALXAC = DSQRT(DARG)

ALXAC = SQRT(AILXAC * AILXAC + AJLXAC * AJLXAC + AKLXAC * AKLXAC)

**THE PRECEDING YIELD THE UNIT NORMALS TO THE FLAT PLATE SEGMENTS**

**AREA DETERMINATION FOR TRAPEZOIDAL SEGMENT OF TWISTED FLAT PLATE**
DELX = CC(L)/NUMM1
AREA = DELX * ALXAC / AC

**UNSTEADY PRESSURE FORCE**
FRCE = SUMP / AREA / ALXAC
FXUS(I,L) = FRCE * AILXAC
FYUS(I,L) = FRCE * AJLXAC
FZUS(I,L) = FRCE * AKLXAC

**RESULTANT VELOCITY OF BLADE CONTROL POINT RELATIVE TO**

**BLADE-FIXED COORDINATE SYSTEM**

VXPOW = RPS * YC(I,L) - (WYC(I,L) * COSTH - WXC(I,L) * SINTH)
VYPOW = -RPS * XC(I,L) - (WXC(I,L) * COSTH + WYC(I,L) * SINTH)
VZPOW = -WZC(I,L)

PUS(I,L) = FXUS(I,L) * VXPOW + FYUS(I,L) * VYPOW + FZUS(I,L) * VZPOW
P(I,L) = PQS(I,L) + PUS(I,L)
FXBV(I,L) = FXQS(I,L) + FXUS(I,L)
FYBV(I,L) = FYQS(I,L) + FYUS(I,L)
FZPV(I,L) = FZQS(I,L) + FZUS(I,L)
CONTINUE
PRINT 1001
CT = Q.
CP=0.
C 400
   CTI=0.
   CPI=0.
   POWER=0.
   THRUST=0.
   DO 90 L=1,NOPAN
      THRST(L)=0.
      POWER(L)=0.
      DRAG(L)=0.
      TCRQ(L)=0.
   C** SPANWISE DISTRIBUTION
   DO 91 L=1,NUMMI
      THRST(L)=THRST(L)+FZBV(I,L)
      POWER(L)=POWER(L)+P(I,L)
      DRAG(L)=DRAG(L)+FXBV(I,L)
      91 TCRQ(L)=DRAG(L)*YC(L)
      DLCTIP=BL*THRST(L)/COEF
      DLCPIP=(BL*POWER(L)/(COEF*VTIP))*PI**4/4.
      CTI=CTI+DLCTIP
      CPI=CPI+DLCPIP
      DLCTIP=DLCTIP/((Y(L+1)-Y(L))/R)
      DLCPIP=DLCPIP/((Y(L+1)-Y(L))/R)
      PHI=ATAN(DRAG(L)/THRST(L))
      DARG=DRAG(L)/THRST(L)
      PHI=DATAN(DARG)
      PRINT 1050,PHI
      DARG=PHI
      SINFI=SIN(PHI)
      COSFI=COS(PHI)
      SINFI=D*SIN(DARG)
      COSFI=D*COS(DARG)
      ALFA=(BETAC(L)-PHI)*180./PI
      IF(THC(L).GE..08) AO=-0.0352*THC(L)+0.1109
      IF(THC(L).GE..21) AO=-0.1525*THC(L)+0.13815
      CL=AO*ALFA
      CDMIN=0.01563*THC(L)+0.004
      CD=CDMIN
      VEVT=YC(L)*COSFI/R
      SIGMA=BL*CC(L)/(PI*R)
      CED DELCTH=VEVT*SIGMA*(CL*COSFI-CD*SIONFI)/2.
      CED DELCPH=VEVT*SIGMA*(CL*SIONFI+CD*COSFI)*YC(L)/R/2.
      CED DELCTP=DELCTH*PI**3/4.
      CED DELCPP=DELCPH*PI**4/4.
      PRINT1009,DLCTIP,DLCPIP,ALFA,DELCTP,DELCPP
      CED CT=CT+DELCTP*(Y(L+1)-Y(L))/R
      CED90 CP=CP+DELCPP*(Y(L+1)-Y(L))/R
   90 CONTINUE
   PRINT 1002
   PRINT1009,CTI,CP,CT,CPI
   CED IF(ITIME.EQ.MXTIME)GO TO 3004
   PRINT 1004
C************************** TGAM FOR NEXT TIME STEP *********************
DC 41 II=1, MATRX1
41 GAM(II)=GAMMA(II)

C*************************** SHELD VORTICES ADDED **************************
L=0
DO 42 II=NUM, MATRX1, NUM
L=L+1
GAMS(L, ITIME+1)=GAMMA(II)
RAS(L, ITIME+1)=GAMS(L, ITIME+1)/(2.0*PI*SUMARR)
C DO104 L=1, NOPAN
C 104 PRINT1003, GAMS(L, ITIME+1)
C PRINT 1004

C************************** CONSERVATION OF ANGULAR MOMENTUM, SHELD **********
IF(LINWA.NE.1) GO TO 1195
DO 43 L=1, NOPAN
GAMS(L, ITIME+1)=GAMS(L, ITIME+1)*SQRT((X(NUM, L+1)—X(NUM, L))**2+(Y(L
L+1)—Y(L))**2+ (Z(NUM, L+1)—Z(NUM, L))**2)
43 CONTINUE
1195 CONTINUE
C DO101 L=1, NOPAN
C 101 PRINT 1003, GAMS(L, ITIME+1)
C PRINT 1004

C************************** STRENGTHS OF TRAILERS ADDED ***********************
SUM1=0.
DO 45 L=1, N0PAN1
M=(L-1)*NUM+1
N=M+NUM-2
SUM2=0.
DO 44 II=M, N
IF(L.EQ.NOPAN+1)GO TO 44
SUM2=SUM2+GAMMA(II)
44 CONTINUE
GAMT(L, ITIME+1)=SUM1-SUM2
RAT(L, ITIME+1)=GAMT(L, ITIME+1)/(2.0*PI*SUMARR)
RAT(L, ITIME+1)=ABS(RAT(L, ITIME+1))
45 SUM1=SUM2
PRINT 7777
DO 102 L=1, N0PAN1
102 PRINT 8888, RAS(L, ITIME+1), RAT(L, ITIME+1)
C IF(ITIME.EQ.MXTIME)GO TO 3004
IF(ITIME.EQ.MXTIME)GO TO 14
C 102 PRINT 1003, GAMS(L, ITIME+1)
C PRINT 1004

C******** WAKE COORDINATE POSITION WRT PROPELLER DISC PLANE
ITIMPL=ITIME+1
DO 50 ITT=1, ITIMPL
IT=ITIME-ITT+2
DO 50 L=1, NOPAN1

C TRAILING EDGE SHEED FILAMENT POSITIONS TRANSFORMED TO PROPELLER
C****************************COORDINATES
XW(L, 1)=X(NUM, L)*COSTH-Y(L)*SINTH
YW(L, 1)=Y(L)*COSTH+X(NUM, L)*SINTH
ZW(L, 1)=Z(NUM, L)
XD=XW(L,IT)
YD=YW(L,IT)
ZD=ZW(L,IT)

C********** VELOCITY DUE TO BOUND VORTICITY **********
VXB=0.
VYB=0.
VZB=0.
IF(LINWA.EQ.1) GO TO 79
COT=COSTH
SIT=SINTH
H=HH
IWAKE=0
CALL INVEL
C IF(KMAX.EQ.NUM) GO TO 1151
C IF(KMAX.EQ.NPANP1) GO TO 1151
C IF(ITT.GT.1) GO TO 1151
C IF(L.GT.1) GO TO 1151
C PRINT 7777
C DO 1137 K3=L,NPANP1
C1137 PRINT 8888,AN(K3),RA(K3)
C1137 PRINT 8888, RAS(K3,ITIME+1),RAT(K3,ITIME+1)
1151 VXW=VXP
VYB=VYP
VZB=VZP
79 CONTINUE

C********** VELOCITY AT WAKE POINTS DUE TO INTERACTION **********
C WAKE COORDINATES IN PROPELLER AXIS SYSTEM
VXW=0.
VYW=0.
VZW=0.
IF(LINWA.EQ.1) GO TO 688
IF(ITIME.EQ.0) GO TO 688
COT=1.
SIT=0.
H=HH
IWAKE=1
CALL INVEL
C IF(KMAX.EQ.NUM) GO TO 1152
C IF(KMAX.EQ.NPANP1) GO TO 1152
C IF(ITT.GT.1) GO TO 1152
C IF(L.GT.1) GO TO 1152
C PRINT 7777
C DO 1135 K4=L,NPANP1
C1135 PRINT 8888,AN(K4),RA(K4)
C1135 PRINT 8888, RAS(K4,ITIME+1),RAT(K4,ITIME+1)
1152 VXW=VXP
VYW=VYP
VZW=VZP

C****** LOCAL SELF-INDUCED VELOCITY **********
ITRAIL=0

C 500
X2=XD
Y2=YD
688  V1S=0.
    VJS=0.
    VKS=0.
    IF(LINWA.EQ.1) GO TO 68
    IF(ITIME.EQ.0) GO TO 68
558  IF(ITRAIL)550,550,551
550  IF(L.EQ.1 OR L.EQ.NPANPL) GO TO 552
     X1=XW(L-1,IT)
     Y1=YW(L-1,IT)
     Z1=ZW(L-1,IT)
     X3=XW(L+1,IT)
     Y3=YW(L+1,IT)
     Z3=ZW(L+1,IT)
     DELS2=SQRT((X3-X2)**2+(Y3-Y2)**2+(Z3-Z2)**2)
     DELS1=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
     AK=(GAMS(L-1,IT)/DELS1+GAMS(L,IT)/DELS2)/2.
     IF(LINWA.EQ.1) GO TO 3000
     AK=(GAMS(L-1,IT)+GAMS(L,IT))/2.
3000  CRB=RAS(NPANPL,IT)=RAS(NPAN,IT)
      GC TO 553
551  IF(IT.EQ.ITIMPL) GO TO 552
C********** END OF WAKE ***********
    IF(IT.EQ.1) GO TO 554
C********** TRAILING EDGE ***********
     X1=XW(L,IT-1)
     Y1=YW(L,IT-1)
     Z1=ZW(L,IT-1)
     DELS1=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
     AKI=GAMT(L,IT)/DELS1
     IF(LINWA.EQ.1) GO TO 3001
     AKI=GAMT(L,IT)
3001  GO TO 555
554  X1=X(NUMM1,L)*COSTH+Y(L)*SINTH
     Y1=Y(L)*COSTH+X(NUMM1,L)*SINTH
     Z1=Z(NUMM1,L)
     DELS1=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
     M=(L-1)*NUM+1
     N=M+NUM-2
C********** TRAILER STRENGTHS (+) FEEDING DOWNSTREAM ***********
     AK1=0.
     DO 237 J=M,N
     IF(L-1) 238,239,238
239  AK1=AK1-GAMMA(J)
     GO TO 237
238  IF(L-NPANPL) 241,240,241
240  AK1=AK1+GAMMA(J-NUM)
     GO TO 237
241  AK1=AK1-GAMMA(J)+GAMMA(J-NUM)
CONTINUE

555  X3=XW(L,IT+1)
       Y3=YW(L,IT+1)
       Z3=ZW(L,IT+1)
       DELS2=SQRT((X3-X2)**2+(Y3-Y2)**2+(Z3-Z2)**2)
       AK2=GAMT(L,ITT-1)/DELS2
       IF(LINWA.EQ.1) GO TO 3002
       AK2=GAMT(L,ITT-1)

3002  CRB=RAT(L,ITT-1)
       AK=(AK1+AK2)/2.

553  CONTINUE
       IF(DELS1.LT.CRB.OR.DELS2.LT.CRB) GO TO 552

      XX=((X3-X2)/DELS2+(X1-X2)/DELS1)/((DELS1+DELS2)/2.)
      YY=((Y3-Y2)/DELS2+(Y1-Y2)/DELS1)/((DELS1+DELS2)/2.)
      ZZ=((Z3-Z2)/DELS2+(Z1-Z2)/DELS1)/((DELS1+DELS2)/2.)
      XXX=((X3-X2)*DELS1/DELS2—(X1—X2)*DELS2/DELS1)/(DELS1+DELS2)
      YYY=((Y3—Y2)*DELS1/DELS2—(Y1—Y2)*DELS2/DELS1)/(DELS1+DELS2)
      ZZZ=((Z3—Z2)*DELS1/DELS2—(Z1—Z2)*DELS2/DELS1)/(DELS1+DELS2)

C      IF(ITRAIL)2200,2200,2201
C2200   PRINT2005
C      GO TO 2202
C2201   PRINT 2006
C2202   CONTINUE
2005 FORMAT(' ' , 'SHED SHED SHED SHED SHED SHED SGED SHED SHED SHED ')
2006 FORMAT(' ' , 'TRAIL TRAIL TRAIL TRAIL TRAIL TRAIL TRAIL TRAIL ')
C      PRINT2000,X3,Y3,Z3,DELS2
C      PRINT2001, X1,Y1,Z1,DELS1
C      PRINT2002, X2,Y2,Z2
C      PRINT2003,XD,YD,ZD
C      PRINT2004,ITT,L,AK,XX,YY,ZZ,XXX,YYY,ZZZ
2000 FORMAT(' ' , 'X3=',E15.8,' Y3=',E15.8,' Z3=',E15.8,' DELS2=',E15.8)
2001 FORMAT(' ' , 'X1=',E15.8,' Y1=',E15.8,' Z1=',E15.8,' DELS1=',E15.8)
2002 FORMAT(' ' , 'X2=',E15.8,' Y2=',E15.8,' Z2=',E15.8)
2003 FORMAT(' ' , 'XD=',E15.8,' YD=',E15.8,' ZD=',E15.8)
2004 FORMAT(' ' , 'Z(14,2X),7(E14.5,2X))
      VIS=VIS+AK*(YYY*ZZ—ZZZ*YY)
      VJS=VJS+AK*(ZZZ*XX—XXX*ZZ)
      VKS=VKS+AK*(XXX*YY—YYY*XX)

552 CONTINUE
       IF(ITRAIL)556,556,557
556  ITRAIL=1
C      PRINT 1004
557  CONTINUE
68  VI(L,IT)=VXB+VXW+V+VIS
   VJ(L,IT)=VYB+VYW+VJS
   VK(L,IT)=VZB+VZW+VKS
C*** INDUCED VELOCITIES AT WAKE POINTS WRT PROPELLER DISC PLANE
PRINT1004
       IF(LINWA.EQ.1) GO TO 74
U6 72  IT=1, ITIMPI
CG 73  L=1,NPANP1
C PRINT1005, IT, XW(L, IT), YW(L, IT), ZW(L, IT), VI(L, IT), VJ(L, IT), VK(L, IT)
C
1, L
   D(1) = XW(L, IT)
   D(2) = YW(L, IT)
   D(3) = ZW(L, IT)
   STATUS(1) = 0
C
73 CONTINUE
72 CONTINUE
   STATUS(1) = 1
   CALL BLKWRIT(5LTAPE1, 3, D, STATUS)
C
74 CONTINUE

C********** CALCULATION OF WAKE COORDINATE POSITION *******************
DO 69 L = 1, NPANPI
   DO 69 IT = 1, ITIMPI
      IT = ITIME - IT + 2
      XW(L, IT + 1) = XW(L, IT) + VI(L, IT) * DELT
      YW(L, IT + 1) = YW(L, IT) + VJ(L, IT) * DELT
      ZW(L, IT + 1) = ZW(L, IT) + VK(L, IT) * DELT
69
C ********** CONSERVATION OF ANGULAR MOMENTUM, TRAILERS **************
   IF (LINWA .NE. 1) GO TO 1196
   DO 70 L = 1, NPANPI
      GAMT(L, ITIME + 1) = GAMT(L, ITIME + 1) * SQRT((XW(L, 2) - XW(L, 1)) ** 2 + (YW(L, 2) - YW(L, 1)) ** 2 + (ZW(L, 2) - ZW(L, 1)) ** 2)
70 CONTINUE
C
1196 CONTINUE
   ITIME = ITIMPI
   PRINT1014, ITIME
   GO TO 71
C3004 PMIN = PMIN + 100.0
CE PRINT 5555
CE PRINT 6666, PMIN
CE IF (PMIN .GT. 1000.0) GO TO 1192
CE GO TO 3003
14 CONTINUE
C1191 CONTINUE
C CALL BLKREWD(5LTAPE1)
1192 STOP
END
SUBROUTINE INVEL
DOUBLE PRECISION DARG
COMMON X(10, 30), Y(30), Z(10, 30), XW(30, 100), YW(30, 100), ZW(30, 100), GAMMA(100), GAMTS(30, 100), GAMTS(30, 100), AN(30), RA(30), CMA(30), CI301, BETAC(30), RAS(30, 100), RAT(30, 100)
COMMON VXP, VYP, VZP, COT, CITIME, IWAKE, BL, IBL, NOPAN, NUM, XD, YD, ZD, IH, E, AO, KMAX, LINWA, V, SUMARR
PI = 3.1415927
VXP = 0.
VYP = 0.
VZP = 0.
IF(IWAKE)1,1,2
2  ITEST=0
   GO TO 23
1  ITEST=-1
23  CONTINUE
   DO 7 IB=1,IBL
   DARG=2.DO*PI*(IB-1)/BL
   CSIN=DCOS(DARG)
   SSIN=DSIN(DARG)
   C
   SSIN=SIN(2.*PI*(IB-1)/BL)
   C
   CSIN=COS(2.*PI*(IB-1)/BL)
   COSBL=COT*CSIN-SIN*SSIN
   SINBL=SIN*CSIN+COT*SSIN
   IF(I)49515
   IF(I)4,5,5
   4  JMAX=NOPAN
   KMAX=NUM
   KK=0
   GO TO 6
   5  JMAX=ITIME
   6  DO 7 J=1,JMAX
      IF(I)8,9,9
      J1=J+1
      J2=JMAX-J+1
      KMAX=NOPAN
      IF(I).GT.0) KMAX=KMAX+1
   8  DO 26 K=1,KMAX
      IF(I)10,11,11
      JJ=KK*NUM+K
      GAM=GAMMA(JJ)
      CRA = H
      IF(K-<NUM)12,13,12
      K1=1
      K2=3
      GO TO 15
      12  K1=2
      K2=2
      GO TO 15
      11  K1=1
      K2=1
      15  DO 26 KKK=K1,K2
         IF(I)29,30,20
      29  GO TO (16,17,18),KKK
      16  XA=X(NUM,J)*COSBL-Y(J)*SINBL
      XB=X(K,J)*COSBL-Y(J)*SINBL
      YA=Y(J)*COSBL+X(NUM,J)*SINBL
      YB=Y(J)*COSBL+X(K,J)*SINBL
      ZA=Z(NUM,J)
      ZB=Z(K,J)
      GO TO 19
      17  XA=X(K,J)*COSBL-Y(J)*SINBL
      XB=X(K,J+1)*COSBL-Y(J+1)*SINBL
      YA=Y(J)*COSBL+X(K,J)*SINBL
      YB=Y(J+1)*COSBL+X(K,J)*SINBL
      GO TO 19
ZA = Z(K, J)
ZB = Z(K, J+1)
GO TO 19

18 XA = X(K, J+1) * COSBL - Y(J+1) * SINBL
XB = X(NUM, J+1) * COSBL - Y(J+1) * SINBL
YA = Y(J+1) * COSBL + X(K, J+1) * SINBL
YB = Y(J+1) * COSBL + X(NUM, J+1) * SINBL
ZA = Z(K, J+1)
ZB = Z(NUM, J+1)
GO TO 19

30 LL = K
LLL = K
II = J
III = J
GAM = GAMS(LL, J2)
CED
RAS(LL, J2) = GAMS(LL, J2) / (2.0 * PI * SUMARR)
CED
RAS(LL, J2) = ABS(RAS(LL, J2))
CRA = RAS(LL, J2)
GO TO 21

20 LL = K
LLL = K
II = J
III = J
GAM = GAMS(LL, J2)
CED
RAT(LL, J2) = GAMS(LL, J2) / (2.0 * PI * SUMARR)
CED
RAT(LL, J2) = ABS(RAT(LL, J2))
CRA = RAT(LL, J2)

21 XA = XW(LL, II) * COSBL - YW(LL, II) * SINBL
XB = XW(LLL, III) * COSBL - YW(LLL, III) * SINBL
YA = YW(LL, II) * COSBL + XW(LL, II) * SINBL
YB = YW(LLL, III) * COSBL + XW(LLL, III) * SINBL
ZA = ZW(LL, II)
ZB = ZW(LLL, III)

19 XBA = XB - XA
C 700

YBA = YB - YA
ZBA = ZB - ZA
XDA = XD - XA
YDA = YD - YA
ZDA = ZD - ZA
XDB = XD - XB
YDB = YD - YB
ZDB = ZD - ZB

ALS = XBA * XBA + YBA * YBA + ZBA * ZBA
ACS = XDA * XDA + YDA * YDA + ZDA * ZDA
BCS = XDB * XDB + YDB * YDB + ZDB * ZDB
DARG = ALS
AL = DSQRT(DARG)
DARG = ACS
AC = DSQRT(DARG)
DARG = BCS
BC = DSQRT(DARG)
C
AL = SQRT(ALS)
C SUBROUTINE TO MULTIPLY TWO MATRICES -- SINGLE PRECISION
C A = VARIABLE NAME OF THE PREMULTIPLIER MATRIX
C B = VARIABLE NAME OF THE POSTMULTIPLIER MATRIX
C C = VARIABLE NAME OF THE PRODUCT MATRIX
C M = NUMBER OF ROWS IN THE PREMULTIPLIER MATRIX
C N = NUMBER OF COLUMNS IN THE PREMULTIPLIER MATRIX
C K = NUMBER OF COLUMNS IN THE POSTMULTIPLIER MATRIX
C JA = NUMBER OF ROWS IN THE PREMULTIPLIER MATRIX AS DIMENSIONED
C JB = NUMBER OF ROWS IN THE POSTMULTIPLIER MATRIX AS DIMENSIONED
C JC = NUMBER OF ROWS IN THE PRODUCT MATRIX AS DIMENSIONED
C
C /* PSUC MXMLT */
C
C ****
C MXMLT
C ****
C
C AC = SORT(ACS)
C BC = SORT(BCS)
AILXAC=YBA*ZDA-ZBA*YDA
AJLXAC=ZBA*XDA-ZDA*XDA
AKLXAC=XBA*YDA-XDA*YBA
1140 IF(IWAKE)31,31,34
34 HH=H
GO TO 32
31 HH=E
32 CONTINUE
IF(AL.LT.CRA) GO TO 26
IF(AC.LT.CRA) GO TO 26
IF(BC.LT.CRA) GO TO 26
COSA=(ACS+ALS-BCS)/(AC+AL*2.)
TEMPA=ABS(1.-COSA*COSA)
DARG=TEMPA
HCORE=AC*DSORT(DARG)
C HCORE=AC*SORT(TEMPA)
IF(HCORE.LE.CRA) GO TO 26
GBS=(BCS+ALS-ACS)/(BC*AL*2.)
IF(IWAKE)24,24,25
IF(IWAKE)24,24,25
25 AL=ALS
24 VFN=GAM*(COSA+COSB)/(AL*ACS*TEMPA*PI*4.)
VXP=VXP+VFN*AILXAC
VYP=VYP+VFN*AJLXAC
VZP=VZP+VFN*AKLXAC
26 CONTINUE
1148 IF(ITEST)27,7,7
27 KK=J
7 CONTINUE
IF(ITEST)28,22,28
22 ITEST=1
GO TO 23
28 RETURN
END
DIMENSION A(JA,N), B(JB,K), C(JC,K)
DO 1 I=1,M
  DO 2 J=1,K
    SUM=0.0
    DO 2 L=1,N
      SUM = SUM + A(I,L)*B(L,J)
  1 C(I,J) = SUM
RETURN
END

SUBROUTINE MXINV(A,MDIM,N)
REAL A(MDIM,N),BIGA,HOLD
INTEGER L(100),M(100)
DO 80 K=1,N
  L(K)=K
  M(K)=K
  BIGA=A(K,K)
  DO 20 J=K,N
    DO 20 I=K,N
      IF(ABS(BIGA)-ABS(A(I,J))) 15,20,20
  15 BIGA=A(I,J)
      L(I)=J
      M(K)=J
  20 CONTINUE
  J=L(K)
  IF(J-K) 35,35,25
  35 DO 30 I=1,N
    HOLD=-A(K,I)
    A(K,I)=A(J,I)
  30 A(J,I)=HOLD
  35 I=M(K)
  IF(I-K) 45,45,38
  38 DO 40 J=1,N
    HOLD=-A(J,K)
    A(J,K)=A(J,I)
  40 A(J,I)=HOLD
  45 DO 55 I=1,N
    IF(I-K) 50,55,50
  50 A(I,K)=A(I,K)/(-BIGA)
  55 CONTINUE
  DO 65 I=1,N
    HOLD=A(I,K)
  65 CONTINUE
RETURN
END
SUBROUTINE GAUSS(N)

COMMON XX(10,30), Y(30), Z(30), XW(30,100), YW(30,100), ZW(30,100), GAMMA(100), GAMS(30,100), GAMT(30,100), X(30), RA(30), OMA(30), C(130), BETA(30), AS(30), A(30,31), YYC(30), CC(30), BETAC(30), RAS(30,100), RAT(30,100)
COMMON VXP, VYP, VZP, COT, SIT, ITIME, IWAKE, BL, IBL, NOPAN, NUM, XD, YD, ZD, H, E, AO, R, KMAX, LINWA, V, SUMARR

C SOLVE A SET OF N SIMULTANEOUS EQUATIONS WITH N UNKNOWNS BY USE
C OF GAUSSIAN ELIMINATION. ***NOTE*** ALWAYS INSURE THAT THE
C DIMENSION STATEMENT IS ALSO REGISTERED IN THE MAIN PROGRAM *****
C TO SET UP THE PROGRAM THE NUMBER OF EQUATIONS IS N THE C65662
C ARE CALCULATED IN THE MAIN PROGRAM AND PLACED IN THE MATRIX A.
C THE PROGRAM SOLVES FOR THE UNKNOWNS X AND RETURNS TO THE MA15
C PROGRAM
NP=N+1
NM=N-1
DO 10 K=1,NM
KP=K+1
L=K
DO 11 I=KP,N
IF (ABS(A(I,K)) .GT. ABS(A(L,K))) L=I
11 CONTINUE
IF (A(L,L) .LT. 0.0) STOP
PRINT, 'YOU CANT DO IT THIS WAY'
STOP
33 IF (L.EQ.K) GO TO 71
DO 70 J=K,NP
TEMP=A(K,J)
A(K,J) = A(L,J)
A(L,J) = TEMP
70 CONTINUE
71 CONTINUE
DO 10 I = KP, N
B = A(I,K) / A(K,K)
DO 10 J = KP, NP
10 A(I,J) = A(I,J) - B * A(K,J)
X(N) = A(N, NP) / A(N, N)
CC 12 IN = 1, NM
I = N - IN
X(I) = A(I, NP)
IP = I + 1
DO 13 J = IP, N
13 X(I) = X(I) - A(I,J) * X(J)
12 X(I) = X(I) / A(I,I)
RETURN
END
```
//DATA_INPUT  DU *

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>10.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.025</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.075</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.125</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.175</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.225</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.275</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.325</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.375</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.425</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.475</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.525</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.575</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.625</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.675</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.725</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.775</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.825</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.875</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.925</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>0.975</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
<tr>
<td>1.00</td>
<td>5.729578</td>
<td>0.0</td>
<td>0.333333330.0</td>
</tr>
</tbody>
</table>
```

Original page is of poor quality.