NASA Technical Paper 1729

WIND - Computer Program for Calculation of Three-Dimensional Potential Compressible Flow About Wind Turbine Rotor Blades

Djordje S. Dulikravich

OCTOBER 1980

NASA

NASA Technical Paper 1729

WIND - Computer Program for Calculation of Three-Dimensional Potential Compressible Flow About Wind Turbine Rotor Blades

Djordje S. Dulikravich
Lewis Research Center
Cleveland, Ohio

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1980

Summary

A computer program (WIND) has been developed that numerically solves an exact, full-potential equation (FPE) for three-dimensional, steady, inviscid flow through an isolated wind turbine rotor. The program automatically generates a three-dimensional, boundary-conforming grid and iteratively solves the FPE while fully accounting for both the rotating cascade and Coriolis effects. The numerical techniques incorporated involve rotated, type-dependent finite differencing, a finite volume method, artificial viscosity in conservative form, and a successive line overrelaxation combined with the sequential grid refinement procedure to accelerate the iterative convergence rate. Consequently WIND is capable of accurately analyzing incompressible and compressible flows, including those that are locally transonic and terminated by weak shocks. WIND can also be used to analyze the flow around isolated aircraft propellers and helicopter rotors in hover as long as the total relative Mach number of the oncoming flow is subsonic.

Introduction

An exact, full-potential equation (FPE) that governs three-dimensional steady, inviscid, potential flow through the wind turbine rotor can be expressed in the following vector form (refs. 1 and 2):

$$a^2\Delta^2\varphi-(\Delta\varphi\cdot\Delta)\frac{1}{2}(\Delta\varphi\cdot\Delta\varphi)+2(\Delta\varphi\cdot\Delta)((\mathbf{\Omega}\times\mathbf{r})\cdot\Delta\varphi)-((\mathbf{\Omega}\times\mathbf{r})\cdot\Delta)((\mathbf{\Omega}\times\mathbf{r})\cdot\Delta\varphi)=0 \qquad (1)$$

where $\mathbf{V}=\Delta\varphi$ is the absolute velocity vector defined as

$$\mathbf{V}=\mathbf{V}_r+\mathbf{\Omega}\times\mathbf{r} \qquad (2)$$

Here $\mathbf{V}_r$ is the relative velocity vector of the fluid with respect to the rotating blade, $\mathbf{\Omega}$ is the angular rotor speed, and $\mathbf{r}$ is the position vector in the rotor plane.

Equation (1) is a second-order, quasi-linear, partial differential equation of mixed elliptic-hyperbolic type. For the particular set of boundary conditions the solution of FPE can be composed of locally hyperbolic regions (supersonic relative flow) that neighbor regions where FPE is locally of an elliptic type (i.e., where the local relative flow is subsonic). The boundary surfaces separating these regions can be of a continuous parabolic type (sonic surfaces) or of a mathematically discontinuous type (isentropic shocks). This means that the FPE can govern flow regimes varying from incompressible (when the FPE reduces to a simple Laplace's equation) through the transonic, accepting mathematical discontinuities in the "weak solution" form. These discontinuities are then interpreted as weak isentropic shocks (refs. 3 and 2).

The FPE cannot be solved analytically, and therefore a numerical method will be used. This method, called the finite-volume method (refs. 4 and 5), represents a combination of finite difference and finite element techniques. Because of the exceedingly large number of mathematical operations that will be performed and the large amount of computer storage required, an iterative, successive line overrelaxation (SLOR) technique is used. The details of the numerical scheme have been presented in other references (refs. 6, 7, and 2) and will not be repeated in this users manual. To make the computer program suitable for computers with a limited central memory, the program was written in such a way that a number of disks or tapes will accommodate the excessive storage requirements.

Applicability of Computer Program WIND

The computer program WIND is capable of numerically analyzing the flow field about a given blade shape of the wind turbine rotor (horizontal-axis type).

The rotor hub is defined as a doubly infinite circular cylinder having the x coordinate as its axis. An arbitrary number of blades can be attached to the hub. The blades can have arbitrary spanwise distribution of taper and of the twist, sweep, and dihedral angles. An arbitrary number of different airfoil section shapes can be used along the span, although the spanwise variation of all the geometric parameters must be reasonably smooth.

The results of the numerical analysis are given in the form of a computer printout. At a number of spanwise stations along the blade the output gives the chordwise distribution of the coefficient of pressure, the Mach number, the density, and the relative velocity vector components. In addition, the results specify local values of the lift coefficient and tangential and axial aerodynamic force components. These are also given in an integrated form expressing the total torque on the shaft and the total axial force on the shaft.

As stated in reference 8, WIND numerically solves an exact, full-potential equation for a three-dimensional, steady, homentropic flow of an inviscid, compressible, homocompositional, nonradiating, and nonconducting fluid. The practical implications of this mathematical model are the following:

(1) The wind is blowing at a steady speed. Gusts lasting for a period of time long enough to change the loading of the blade significantly should be analyzed as separate test cases at different wind speeds, each speed treated as a different constant.

(2) The axis of the rotor is always alined with the wind.

(3) The atmospheric turbulence is of negligible intensity.

(4) The tower and all other obstacles are placed downwind from the rotor.

(5) The entire rotor is placed above the ground shear layer; that is, the wind speed is uniform.

(6) Vertical and horizontal variations of atmospheric density and temperature are negligible.

(7) The rotor rotates at constant angular speed.

(8) There is no separation of the airstream from the blades or the rotor hub.

(9) The rotor hub has the shape of an infinitely long, circular cylinder.

(10) Atmospheric impurities (rain, snow, ice, sand, dust, and industrial air pollutants) are uniformly distributed throughout the air mass.

It is important to explicitly warn the users of the WIND code that this program cannot account for possible flow separations and subsequent heavy losses due to poor aerodynamic design of rotor blades.

In its present form WIND is capable of calculating flow fields that could become locally supersonic and terminate with weak shocks. Although this situation is not desirable, it can easily occur in the case of a strong gust or a malfunction of the blade twisting (pitching) mechanism. The consequences include an unacceptably high level of noise and possibly serious damage to the blade and tower structure as the result of an aerodynamic load that was not accounted for during the blade design process.

With increasing rotor diameter and angular speed the outer portions of the blades will operate in a flow regime where the compressibility effects cannot be neglected. The compressibility effects, combined with blade shapes incorporating advanced shockless airfoil shapes, are capable of significantly increasing the aerodynamic efficiency of the rotor without increasing the noise level. The theories for this approach have already been developed (refs. 9 and 10) and successfully tested numerically (ref. 11).

Besides being capable of calculating flows ranging from the incompressible through the transonic regime, WIND fully accounts for the effect of the Coriolis force and the mutual blade interference (cascade) effect. Compressibility and Coriolis effects will be important in the outer portion of the blade; the three-dimensional rotating cascade effects become important in the blade root region.

Global Structure of Program

A computer program called WIND numerically solves the FPE. The program is written in FORTRAN IV and consists of about 2200 computer cards. Wind automatically generates the computational grids and performs the iterative solution of the FPE. Thus user intervention is unnecessary once the program has begun execution.

While developing and testing the program it was observed that the computational grid-generating portion of the code uses about 50 percent of the total high-speed memory required by WIND but consumes less than 2 percent of the total central processing unit time required by WIND. Therefore, to save computer storage and at the same time provide the designer with an opportunity to separately analyze the geometry, WIND was divided into two separate programs (fig. 1).

The first portion of this code is called WIND-01. It reads its own input data, which contain all major geometric parameters. WIND-01 then generates the three-dimensional, body-fitted grid and stores it on disks or tapes.

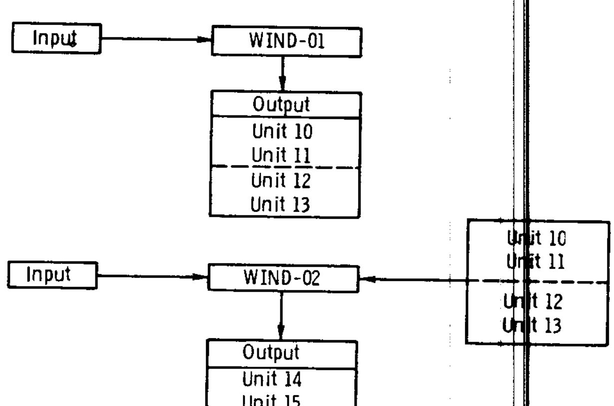


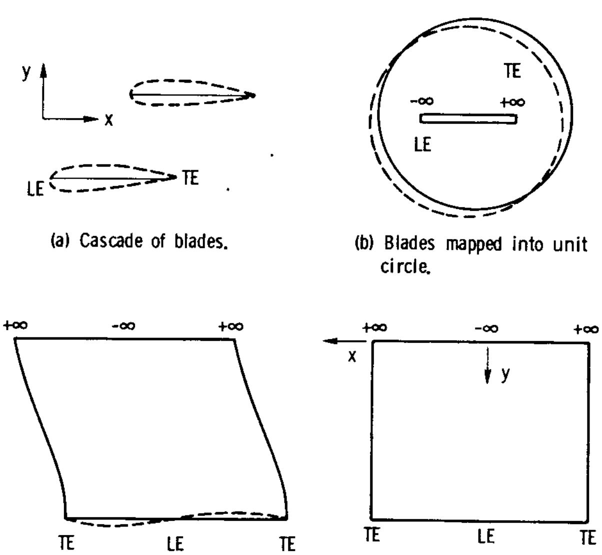
Figure 1. - Global flow chart.

The second portion of the program is called WIND-02. It reads the (x,y,z) coordinates that were previously generated by WIND-01 and thus reactivates units 10, 11, 12, and 13. WIND-02 also reads its own input, which defines the basic flow parameters, and then performs an interative solution of the FPE. The final results of WIND are thus the results of WIND-02. These results, representing values of the potential function, will remain stored on units 14 and 15.

Computer Program WIND-01

The grid generation performed by WIND-01 is based on an analytic function (refs. 12 and 2) that conformally maps a cascade of straight slits onto a cascade of unit circles with a slit in the middle (fig. 2). These circles are "unwrapped" by using elliptic polar coordinates. After additional coordinate stretching and shearing this deformed rhomboid is converted into a parallelepiped-shaped computational domain suitable for finite differencing. Several important geometric parameters and (x,y,z) coordinates of the coarse (first) grid are stored permanently on units 10 and 11, coordinates of the refined (second) grid are stored on unit 12, and coordinates of the fine (third) grid are stored on unit 13 so that each of them can be separately plotted and analyzed.

The computer program WIND-01 consists of seven routines and a separate input data operation (fig. 3). The input data are discussed in detail in the following section. The subroutine MAIN is the principal part of WIND-01 in the sense that most of the other routines are called from that routine. MAIN reads the input data and rotates the airfoil to its actual stagger angle. MAIN also writes (x,y,z) coordinates of the first (coarse) grid on unit 11 and the values of the parameters XCELL, YCELL, ZCELL, RHUB RTIP, RINF, TIPCEL, and BLADE (explained in the following section) on unit 10. On the same unit 10, MAIN also writes the values of the normalized radii of the cylindrical computational surfaces (ref. 8) intersecting the blade. If the original coarse grid has to be refined in all three directions, the refined grid coordinates will automatically be written onto unit 12. The coordinates of the fine (third) grid will be written by MAIN on unit 13.



(a) Cascade of blades.

(b) Blades mapped into unit circle.

(c) Deformed rhomboid.

(d) Parallelepiped.

Figure 2. - Transformation from (x,θ) plane into computational plane, where TE denotes trailing edge and LE denotes leading edge.

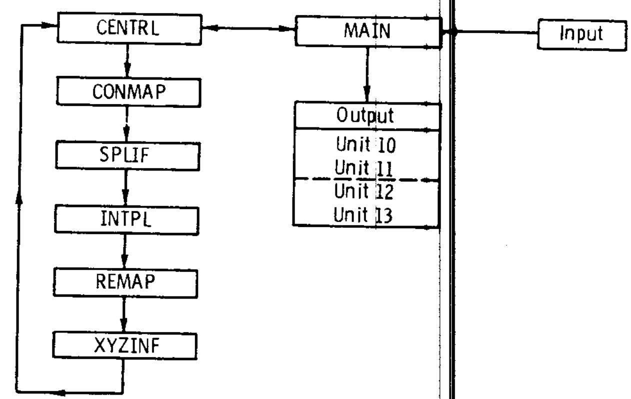


Figure 3. - Flow chart of WIND-01 computer program.

Subroutine CENTRL calculates the length of the central slit in the circle plane and calls routines SPLIF, INTPL, CONMAP, and REMAP in order to perform the conformal mapping and remapping.

Subroutine CONMAP iteratively performs the point-by-point conformal mapping from the (x,θ) plane onto the circle plane. Furthermore CONMAP "unwraps" the circle and calculates the elliptic polar coordinates (fig. 2).

Subroutine SPLIF fits a cubic spline through the lower boundary of the computational domain that corresponds to the surface of the airfoil.

Subroutine INTPL interpolates the values of the elliptic polar coordinates at points that are equidistantly spaced with respect to the image of the upstream infinity. This is a necessary step in obtaining a grid that is periodic in the θ-direction.

Subroutine REMAP analytically determines coordinates of the mesh points in the physical plane; that is, REMAP performs a backtransformation process from the computational plane to the (x,θ) plane.

The mesh points defining the axial infinities should be positioned along the line $x=$constant (fig. 4). Because of the way the potential jump Γ will be enforced across the cut, the points at $x=\pm\infty$ should be equidistantly positioned in the θ direction with respect to that cut.

Subroutine XYZINF determines the coordinates of the axial infinity points explicitly, because they cannot be obtained from the conformal mapping. XYZINF also determines the coordinates of points in the imaginary rows and columns outside the actual computational domain.

Input to WIND-01

All the input data for WIND-01 are combined in one data set (see appendix A). Following is a card-by-card description of that data set (see appendix B for the input format).

The first card contains an arbitrary text composed of up to 80 alphanumeric characters. Its purpose is to give a description or a name to the entire input defining the blade geometry. This same text will then appear at the beginning of the output of WIND-01.

The second card contains the following five parameters:

XCELL — number of computational mesh cells around surface of each blade cross section (fig. 4) when calculating on coarse (first) grid. XCELL must be an even number. The maximum suggested value is XCELL = 24.

YCELL — number of elliptic layers of mesh cells enveloping each local blade cross section (fig. 4) when calculating on a coarse grid. The suggested minimum value is YCELL = 5.

ZCELL — number of mesh cells in spanwise direction (between hub and radial infinity cutoff boundary) when calculating on coarse grid. The minimum value for ZCELL that will still provide for a stable numerical scheme can be determined from the following expression:

$$\text{ZCELL} = 0.9\left(\frac{r_t - r_h}{c_t + c_h}\right) + 2.$$

where r_t is the rotor radius, r_h is the radius of the rotor hub, c_h and c_t are the blade chord lengths at the hub and the tip, respectively.

TIPCEL — number of mesh cells in spanwise direction (between hub and the tip of blade) when calculating on coarse grid. The maximum value for TIPCEL should be

$$\text{TIPCEL} = \text{ZCELL} - 2.$$

PMESH — total number of consecutively refined computational grids that should be generated and separately stored. The number of the mesh cells on the first (coarse) three-dimensional grid is defined by specifying XCELL, YCELL, ZCELL, and TIPCEL. Each following grid will be automatically generated with twice as many mesh cells in each of the computational directions (X,Y,Z) as the previous grid had. The minimum value for PMESH is 1 and the maximum value is 3.

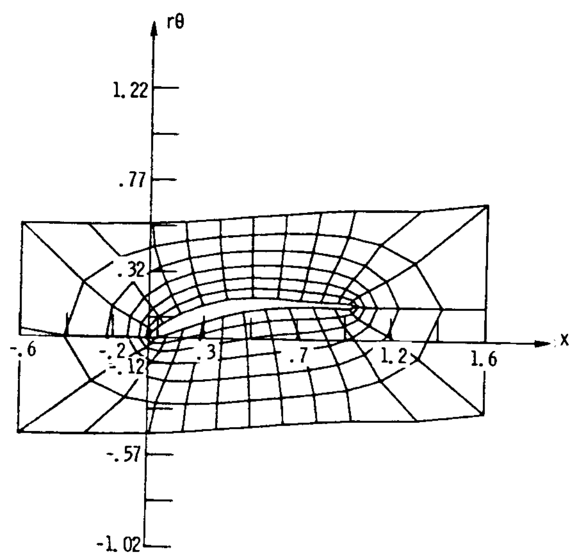


Figure 4. - Computational mesh on (x, rθ) cylindrical surface.

The third input card contains the following five parameters:

RINF — normalized radius of cylindrical outer boundary on which radial infinity boundary conditions will be applied (ref. 8). Normalization must be performed with respect to the rotor radius r_t (fig. 8).

RHUB — normalized radius of rotor hub. Again, the normalizing length is r_t.

RTIP — rotor radius r_t, ft

BLADE — number of rotor blades

SETANG — blade setting angle, deg. This is the angle between the pitching position of the blade as defined by the rest of the input data and any other blade pitching position (fig. 5).

The actual detailed shape of the blade without the hub is given on a number of (x,y) input planes (fig. 6). The innermost input plane must be positioned inside the hub; the outermost input plane should be positioned precisely at or beyond the blade tip.

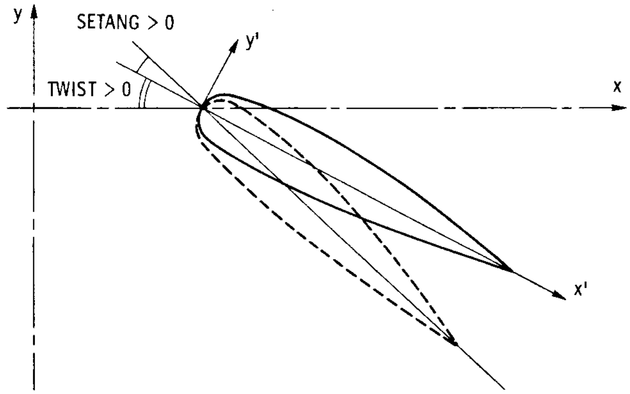


Figure 5. - Local blade section inclination.

The fourth input card contains eight parameters defining the relative position in space of the N^{th} local blade cross section. These parameters are

N — number of (x,y) input plane. The planes are numbered in the spanwise direction, starting with N = 1 for the most inward plane (fig. 6). The maximum number of input planes N = NP is 25.

RLEAD — radial distance from axis of rotation to leading edge of local airfoil defined on N^{th} plane. RLEAD is nondimensional; the normalizing length being the rotor radius RTIP.

CHORD — chord length of blade local cross section defined on N^{th} input plane. Chord length is defined as a maximum distance connecting the leading and trailing edges of the blade section. CHORD is nondimensionalized with respect to RTIP.

XLEAD — axial distance (sweep) from origin of (x,y,z) coordinate system to leading edge of the blade cross section defined on N^{th} plane. The origin should be placed on the axis of rotation (x axis) close to the point of intersection of an extrapolated leading edge and that axis (fig. 6). XLEAD is nondimensionalized with respect to the RTIP.

R01,R02 — leading- and trailing-edge radii of blade local cross-section (airfoil) shape defined in N^{th} plane. R01 and R02 are nondimensionalized with respect to the actual local blade chord length on the N^{th} plane.

TLEAD — local dihedral angle between z axis and leading edge of blade cross section on

The fifth input card, together with a number of following input cards, specifies pairs of (x',y') coordinates of the finite number of input points defining the shape of a local blade cross section on the N^{th} plane. These cards also specify the number of each point and a control parameter (explained below) assigned to each input point. Thus the input parameters appearing on the fifth card (and several consecutive input cards) are (appendix A):

I — number of particular point on local airfoil surface at N^{th} station. The input points are numbered starting from the trailing edge (I = 1) and proceeding clockwise until the trailing-edge point is reached again (I = MAXP). The maximum number of input points at each N^{th} station is MAXP = 65. By increasing the length of vector arrays in COMMON/BLK1/ and COMMON/BLK2/ of the WIND-01 code, the value of MAXP can be increased accordingly.

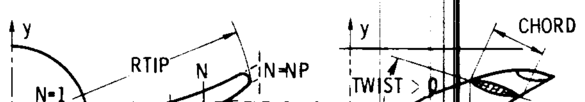
x′ — x' coordinate of I^{th} input point on blade cross section at N^{th} spanwise station. The value of x′ is normalized with respect to the local blade chord length. The range of possible x′ values is

$$0 < x' < 1.$$

y′ — y' coordinate of I^{th} input point on blade cross section at N^{th} spanwise station. The value of y′ is normalized with respect to the local blade chord length.

It is emphasized that the (x',y') input system differs from the (x,y,z) coordinate system (fig. 5). The (x',y') system is attached to the local leading edge of the N^{th} blade cross section, with the x' axis passing through the blade leading- and trailing-edge points. The (x,y,z) system has its origin on the axis of rotation (x axis) and rotates with the blade.

As already mentioned, the fifth card and each number of the following input cards specify x' and y' coordinates of two input points (appendix A). The total number of input points on each N^{th} plane must be odd (counting the trailing-edge point twice, i.e.,



accounting for it as I = 1 and I = MAXP) if the local airfoil shape is not symmetrical. If the local input airfoil shape is symmetrical, the total number of input points must be even, and only the points along the lower airfoil surface need be specified. WIND-01 will then automatically determine the symmetrically positioned remaining input points that are on the upper blade surface.

From appendix A it can be seen that after each pair of (x',y') coordinates specifying an input point on the N^{th} station, there is a three-digit parameter. Its description follows:

NEXT — input reading parameter. The value NEXT = 111 at the end of each input card specifying (x',y') input coordinates means that the following input card has the same format and that the reading of input data of the same type can be continued. The value NEXT = 000 acknowledges the end of the set of input cards specifying (x',y') coordinates and consequently the end of all the data that are needed at the N^{th} station. Such a set involves input card 4 and all the following cards, including a card with NEXT = 000. A value of NEXT = 222 at the end of such a set of cards instead of NEXT = 000 means that the airfoil shape at the $(N+1)^{th}$ station will be the same as it was at the N^{th} station.

An entire set of (x',y') input coordinates has to be formed for each of the N spanwise input stations. Only when the last parameter on the N^{th} station input set is NEXT = 222 will the set for the $(N+1)^{th}$ station consist of a single input card having the format of the fourth input card.

The very last card in the input deck for the WIND-01 code is the same as card 4, but with N specified as N = 0. The rest of the parameters on that card are arbitrary (appendix A).

Output of WIND-01

The results of the WIND-01 program are partially given in written form on a single page of standard computer output (appendix C). This output lists the name of the programmer, the name of the program, and the institution where it was developed. These statements are followed by a listing of major geometric input parameters defining each N^{th} spanwise station. This listing is helpful in debugging the input deck. The next portion of printed output gives the following parameters, obtained after the spanwise interpolation has been performed:

K — number of cylindrical computational surface (fig. 8)

RADII — radii of K^{th} cylindrical computational surface divided by rotor radius (RTIP).

TWIST — twist angle of intersection contour (profile) obtained by intersecting blade with K^{th} cylindrical computational surface, deg

CHORD — chord length of the contour at K^{th} station. CHORD is normalized with respect to RTIP.

XL — axial distance between origin of (x,y,z) coordinate system and leading edge of contour on K^{th} cylindrical surface. XL is normalized with respect to RTIP.

TL — tangential distance between x axis and leading edge of contour on K^{th} cylindrical surface, deg

The major portion of the output from WIND-01 will be permanently stored on disks or tapes (fig. 3). For example, the parameters XCELL, YCELL, ZCELL, RHUB, RTIP, RINF, TIPCEL, and BLADE and the normalized radii of the K computational cylindrical surfaces will be stored on unit 10. The (x,y,z) coordinates defining every mesh point on the coarse three-dimensional grid will always be stored on unit 11. The (x,y,z) coordinates of the next (refined) three-dimensional grid will be automatically defined and stored on unit 12 (if PMESH = 2). If PMESH = 3 in the input data, the (x,y,z) coordinates of the fine (third) grid will be automatically generated and stored on unit 13. A message will appear on the printout sheet each time a separate storage device has been successfully used.

Computer Program WIND-02

The second part of the program WIND, called WIND-02, is executed separately after the (x,y,z) coordinates of the body-conforming grid have been generated by the first part, WIND-01. The purpose of WIND-02 is to actually solve the exact full-potential equation and to calculate important aerodynamic parameters characterizing the flow field. WIND-02 was written in a modular form, separating the main steps in the solution process of the governing equation (fig. 7).

The main program, which reads input data that are written on cards as well as the (x,y,z) coordinates of the computational grid generated by WIND-01, is called MAIN. The same routine calls other routines and refines the potential field after the iteration process on a particular mesh has converged (when PMESH is greater than 1).