NASA Contractor Report 178374

An Engineering Study of Hybrid Adaptation of Wind Tunnel Walls for Three Dimensional Testing

(NASA-CR-178374) AN ENGINEERING STUDY OF HYBRID ADAPTATION OF WIND TUNNEL WALLS FOR THREE-DIMENSIONAL TESTING Final Report (Tracor Hydronautics) 99 p CSCL 01A

N88-14957

Unclas

G3/02 0103584

C.E. Brown, K.M. Kalumuck, and D.N. Waxman

Tracor Hydronautics, Inc.
Laurel, MD. 20707

Contract NAS1-18184
December 1987

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

TABLE OF CONTENTS

Page

1.0 INTRODUCTION 1

1.1 Background 1

1.2 Scope of this Report 3

2.0 ESTIMATION OF TUNNEL FLOW DISTORTION AT MODEL 5

2.1 Tracor Algorithm for Reduction of Blockage 5

2.2 Derivation of PHIXZM - A Green's Source Code for Estimation of Residual Tunnel Flow Distortion 7

2.3 A Simplified AEDC Model Flow-Field Code 9

2.4 Extension to Transonic Nonlinear Wall-Free Codes 10

2.5 Comparison of Green's Source Method with Closed Form Solutions to Assess Numerical Accuracy 11

3.0 APPLICATION OF THE CODES TO THE AEDC MODEL IN THE LANGLEY 0.3 METER TCT FLEXWALL TUNNEL 13

3.1 Basic Model Results 13

3.2 Sensitivity to Span Load Shift 16

3.3 Sensitivity to Fore and Aft Load Shift 17

4.0 CONSIDERATION OF THREE DIMENSIONAL FLEXING OF THE UPPER AND LOWER WALLS 18

5.0 ESTIMATION OF FLOW DISTORTION FROM TEST DATA 20

6.0 CONCLUSIONS 26

7.0 REFERENCES 27

FIGURES 29

APPENDIX A - PHIXZM FORTRAN LISTING 34
APPENDIX B - AFMODL FORTRAN LISTING 40
APPENDIX C - NONLVN FORTRAN LISTING 50
APPENDIX D - JACK_DISPL ANALYSIS AND FORTRAN LISTING 54
APPENDIX E - VEEXPHINO FORTRAN LISTING 87

1.0 INTRODUCTION

1.1 Background

Interference from the presence of wind tunnel walls in model test data has been studied for many years, and corrections have been devised for the influence of solid and straight walls in works by Glauert (Reference 1) and Theodorsen (Reference 2). More recently the idea of adaptable walls that flex to conform to the wall-free stream lines about the model has been pursued actively both in the USA and in Europe (References 3 - 7). See also the bibliography prepared by Tuttle and Mineck (Reference 8). The general flexing of the walls of a wind tunnel introduces a complex mechanical problem and efforts have been made to find simple but still effective ways to reduce residual wall interferences to negligible values. An innovative scheme using solid side walls and flexing rods for upper and lower walls was described by Harney (Reference 9) at the Air Force Wright Aeronautical Laboratory. This report discusses in some detail the problem of proper choice of contouring for hybrid adaptation. At the NASA Ames Research Center, Shairer and Mendosa (Reference 10) describe research in which controlled air flow through porous walls is used in lieu of flexible walls to create the proper outer streamline shape. In England, at the University of Southhampton, a technique was described (Reference 11) in which only the upper and lower walls of the tunnel were made flexible, resulting in a "hybrid adaptation." In this work the influence coefficients of each of the positioning jacks was to be determined experimentally and these data input to a computer for estimation of best wall positioning. At the Arnold Engineering Development Center of the U.S. Air Force, a segmented variable porosity scheme has been employed using sixty-four individually controlled segments (Reference 12).

Smith (Reference 13) has investigated the case of wind tunnels with solid side walls and 2-D flexing upper and lower walls. He presents a method for shaping walls that sharply reduces the model centerline upwash interference and also reduces the axial gradients.

The present report also deals with solid wall tunnels having only upper and lower walls flexing. An algorithm for selecting the wall contours for both two and three dimensional wall flexure is presented and numerical experiments are used to validate its applicability to the general test case of three dimensional lifting aircraft models in rectangular cross-section wind tunnels. The method requires an initial approximate representation of the model flow field at a given lift with walls absent. This representation should be at best a solution of the non-linear transonic flow equations to allow use of the method up to Mach numbers where wall speeds approach unity. The numerical methods utilized are derived by use of so-called Green's source solutions obtained using the method of images. First order linearized flow theory is employed with Prandtl-Glauert compressibility transformations. In addition to two dimensional flexing of the upper and lower walls, three dimensional flexing is also considered for cases in which three rows of jacks could be used or in which only a single centerline row of jacks would be used with fixed unclamped side edges. Equations are derived for the flexed shape of a simple constant thickness plate wall under the influence of a finite number of jacks in an axial row along the plate centerline. As a final task, the Green's source methods are developed to provide estimations of residual flow distortion (interferences) with measured wall pressures and wall flow inclinations as inputs.

1.2 Scope of this Report

This report presents the following Fortran codes developed using a VAX /VMS operating system and essentially incorporates the Fortran 77 notations and conveniences:

(1) PHIXZM - Green's source representation of flow distortion at the model location in the wind tunnel produced by a model flow field in combination with wall boundary shapes. Flow field input can be from numerical computations or from computations based on measured tunnel wall values of pressure and slope.

(2) AFMODL - An approximate representation of an AEDC model, see Reference 13, page 109, using pointed horseshoe vortices for lift of sweptback wings, doublets on the fuselage axis to represent fuselage lift, swept source and sink lines on the wings to represent thickness and sources on the axis for fuselage and balance sting thickness. It also applies the Tracor blockage algorithm and provides an input file for PHIXZM to estimate residual flow distortion at the model due to the hybrid (incomplete) adaptation.

(3) NONLVN - Applies Tracor blockage algorithm using input data from a non-linear code and provides an output for introduction in PHIXZM.

(4) VEEXPHINO - Computes the wall-free normal velocity field at the walls from measured wall slope and pressure. (Wall Slope may result from flexure or boundary layer growth). Provides input to PHIXZM for estimating residual flow distortion at the model region.

(5) JACK_DISPL - Computes the jack displacements and resulting residual normal velocities at the wall for 3-D flexible plate upper and lower walls controlled by a single control row of jacks. The Tracor Algorithm is applied for input into PHIXZM.

A modified version of AFMODL called AFMODLJ is appended to provide input of wall-free normal velocities at the panels and at jack stations for the 0.3 Meter Tunnel.

2.0 ESTIMATION OF TUNNEL FLOW DISTORTION AT MODEL

2.1 Tracor Algorithm for Reduction of Blockage

The presence of fixed straight walls in a rectangular tunnel can be represented in inviscid flow by a doubly infinite set of images of the model as described clearly in Reference 1. This is general and true even for transonic and supersonic flow. However, when the model image fields produce sizable pressure gradients and flow angularity at the model position it will clearly introduce errors in drag, moments, and lift forces that require correction. The ideal case represented by flexible walls contoured to the free air streamlines would permit testing with no corrections for walls and having only Reynolds number mismatches to be corrected for. The boundary layer growth on the walls must of course be included in correctly positioning the walls. Harney (Reference 9) and Wolf, et. al (Reference 11) have shown the possibility of minimizing tunnel flow field distortion by flexing only the upper and lower walls leaving the side walls straight and thereby reducing mechanical complexity and cost of the wind tunnel installation. In both references cited, however, no clear scheme for positioning the flexed 2-D walls was demonstrated for the general case of an aircraft model having both thickness and lift. The Tracor algorithm to be described does in fact show remarkable ability to negate the axial pressure gradient and the upwash on the centerline of the tunnel and to reduce markedly the spanwise upwash variation (washin) normally associated with flat side walls. The algorithm concept is simple. Consider Figure 1 showing an axial view of a lifting model in the tunnel with the streamline traces projected in a plane normal to the axis of the tunnel.

The sketch on the left shows the wall free conditions and that on the right the fixed wall condition. In the lower half of the tunnel there is for the case shown a lack of freedom for the flow to expand beyond the walls hence a blockage exists which requires the flow to speed up abnormally in that section to satisfy continuity of mass flow in the tunnel. At the same time there is a crowding of upward flowing streamline traces near the wing tip leading to an induced upwash over the wing. In Figure 2 where the walls are permitted to effectively move outward as one moves axially the streamline traces can resume their more nearly free air patterns near the model as illustrated in the sketch. A logical manner to determine the wall slope thus appears to be as follows: allow the wall in each half of the tunnel (upper and lower) to permit a net outflow through the boundary equal to the integral of the free-air outflow in the half tunnel. This is essentially the concept of the Tracor Algorithm treated in this report. It requires, of course, a free air estimation of the normal velocity components at the wall control surfaces. It also appears logical that a non-linear transonicfree-air solution at a given lift coefficient can be effectively used so long as the disturbance velocities at the wall are truly small. It is also clear that the lift coefficient of the computed data and the lift coefficient measured or set experimentally should be matched as the lift is the primary driver of the outer flow field independent to first order of its distribution. As will be shown, application of the Tracor algorithm for setting the flex walls reduces the axial pressure gradients to negligible levels. It also sharply reduces the upwash at the model centerline and reduces the lateral upwash gradients to generally acceptable levels.

2.2 Derivation of PHIXZM - A Green's Source Code for Estimation of Residual Tunnel Flow Distortion

Once the residual normal velocities at the tunnel walls (control surface) are given from a free-air calculation and application of the Tracor algorithm, input into PHIXZM yields the tunnel x- and z-wise disturbance velocities near the model. PHIXZM utilizes a Green's source concept; that is, a normal wall velocity can be cancelled over a small panel by a solution of the governing fluid differential equations that produces a uniform normal velocity over the panel and zero normal velocity at all other points on the tunnel walls. The derivation of this Green's source solution is done using the method of images for a tunnel of rectangular cross section and infinitely long in the axial direction. The image system for a Green's source on a wall panel is illustrated in Figure 3. A similar figure of paired sources can be drawn for sources on the floor. Note that symmetry about the centerline is assumed throughout this report. The paneling system used is as follows: an even number of panels is taken vertically on the wall, the wall panels are square but the floor panels can be slightly rectangular depending on the tunnel width to height ratio. The program is written to make them as square as possible. Distances to panel centers in the X direction (downstream) are governed by an integer index, IX, upstream values of IX are negative. Panel coordinates on the walls at a given IX location are made dependent on an integer index, IZ, starting at 1 just to the side of the centerline on the tunnel floor. Thus, panel center coordinates can be specified by two indices IX and IZ. The coordinate system and numbering system are shown in Figure 4. Positions of image sources are defined by indices IZ, IX, and I and L, I being the number of image pairs away from the tunnel vertically and L the corresponding index laterally. At a value of I and L equal to M (arbitrary integer) the calculations are hastened by replacing the outer discrete sources by smearing them uniformly in strength to infinity on the plane being used. This permits the effect of all the additional sources to be integrated in closed form. If the value of M selected is sufficiently large the variation of disturbance velocities across the tunnel by the outer sources is small enough so that both axial and upwash velocities need only be computed at the tunnel center, y & z equal to zero.

The equations for the axial and vertical velocities at the centerline for any sources are as follows:

$$\phi_z(\text{upwash velocity}) = \frac{Q}{4\pi\beta}\left\{\frac{z_1 - z}{[(x_1-x)^2/\beta^2+(y_1-y)^2+(z_1-z)^2]^{3/2}}\right\} \qquad (1)$$

$$\phi_x(\text{axial velocity}) = \frac{-Q}{4\pi\beta^3}\left\{\frac{x_1 - x}{[(x_1-x)^2/\beta^2+(y_1-y)^2+(z_1-z)^2]^{3/2}}\right\} \qquad (2)$$

where Q is the source strength, β is $\sqrt{1 - (\text{Mach})^2}$, x_1, y_1 z_1, and x, y, z are the coordinates of the source and the field point, respectively. These disturbance velocities are derived from the potential function, ϕ, the unit source solution to the linear first order compressible differential equations of motion. The source strength Q, is related to the normal velocity, VN, at its panel by the relation

$$Q = 2\cdot VN\cdot\Delta x\cdot\Delta z \text{ or } 2\cdot VN\cdot\Delta x\cdot\Delta y \qquad (3)$$

for wall or floor panel locations. $\Delta X \Delta Z$ is the area of panel. Making use of the above considerations the program PHIXZM was written to provide ϕ_x and ϕ_z at points at and near the test model with input normal velocity distributions obtained either from estimates (linear or nonlinear) for the model under test at a given lift coefficient, or from the estimated wall-free values calculated by the program VEEXPHINO using measured test values of axial and normal velocities at the wall control surfaces. Fortran listing of the program PHIXZM is given in Appendix A.

2.3 A Simplified AEDC Model Flow-Field Code

In order to exercise the PHIXZM code and assess the beneficial effect of flexwalls set according to the Tracor Algorithm, a wall-free flow field computation was needed. The following describes a simplified modeling approach applied to the AEDC Wind Tunnel Model described in Reference 12. The model itself is simple consisting only of a circular body, with swept back wings and horizontal tail. The body thickness was represented by a single source located behind the nose and a sink located at the discontinuous base sting intersection. The body lift was represented by two semi-infinite doublet lines one originating at the body source and the second (negative) at the body sink. Wing and tail thickness were represented by swept source and sink lines lying respectively at the leading and trailing edges of the airfoils. Wing and tail lift were represented by a swept line vortex, in effect a pointed horseshoe vortex. For a given total lift, the angle of attack was estimated from simple swept wing theory and the main wing was estimated to contribute 80% of the total lift, the tail 20%. The model was assumed to be mounted so that it moves upward with angle of attack on an arc about the center of rotation located well behind the model. The equations for the flow are given in a Fortran listing of the code named AFMODL presented in Appendix B. This program also applies the Tracor Algorithm and produces a file of the displacements of upper and lower surfaces for the Langley 0.3-meter TCT with flexwalls for the AEDC model at arbitrary lift coefficient, CL, and Mach number. This file is called ZDISPL.DAT. The program also produces a file called PHINWALL.DAT containing the estimated wall-free normal velocities at the center of each wall panel for input into PHIXZM.

2.4 Extension to Transonic Nonlinear Wall-Free Codes

The use of small disturbance-linearized theory limits the applicability of the method to moderate combinations of Mach number, model/tunnel size ratio, and lift coefficient. Some extension can be obtained by using a nonlinear transonic code for computing the wall-free flow field up to the condition for which wall speeds approach the speed of sound or deviate substantially from the main flow Mach number. The code used should provide a file called VNZERONL.DAT that can be introduced into a modified version of AFMODL to replace the simplified model representation calculation of the wall-free normal velocities. This has been done and is presented in Appendix C as NONLVN. This program generates a file PHINWALL.DAT for input to PHIXM and a file called ZDISPL.DAT giving the coordinates of the upper and lower walls as dictated by the Tracor blockage algorithm.

2.5 Comparison of Green's Source Method with Closed Form Solutions to Assess Numerical Accuracy

The PHIXZM code for estimating wall induced flow distortion at the model location was checked for accuracy by comparison with Glauert's (Reference 1) nearly closed form equation for the upwash produced by a horseshoe vortex in a square tunnel, namely:

$$\text{Upwash at the tunnel center} = 0.137\, SC_L/D^2 \qquad (4)$$

Here S is the wing area, C_L the lift coefficient and D^2 the tunnel cross sectional area. For the comparison, the program AFMODL was modified to produce a horseshoe vortex by setting sweep, fuselage diameter, wing and tail thickness and tail lift, all to zero. The wall-free normal velocities obtained were then input to PHIXZM to obtain the tunnel center upwash. For a ratio of S/D^2 of 0.021615 and C_L = 0.5, the Glauert value was 0.0014806 and the Green's Source Method 0.0014799. A total of 1640 double panels was used for this computation. To check the accuracy of the axial velocity computations, a direct calculation was made of the axial velocity distribution in the tunnel center produced by a unit strength source at the origin. This numerical solution was calculated using the method of images in a square tunnel. For the comparison, the AFMODL program was modified to produce a unit source by setting lift coefficient, wing and tail thicknesses and rear end sink strength all to zero. Once again the wall-free normal velocities were input to PHIXZM to obtain the axial disturbance velocity ratios due to the walls. The following table shows the result:

Table 1

Axial Disturbance Velocity Ratios

X/D	PHIXZM	DIRECT SOURCE
0.1	0.07137	0.07129
0.2	0.13928	0.13912
0.3	0.20095	0.20072
0.4	0.25470	0.25441
0.5	0.29999	0.29964

Again 1640 double panels were used. Clearly the numerical accuracy is more than adequate considering the general approximations in the basic theory.

3.0 APPLICATION OF THE CODES TO THE AEDC MODEL IN THE LANGLEY 0.3-METER TCT FLEXWALL TUNNEL

3.1 Basic Model Results

The AFMODL program was used to compute the wall-free normal velocities for the AEDC model at a lift coefficient of 0.55 and Mach number of 0.77. The input of these data into PHIXZM after modification by the Tracor blockage algorithm provides an estimate of the residual flow field distortion given in the following table computed using 3240 panel pairs:

Table 2

NX= 40 NZ= 20 MACH=0.770 A=0.330 D=0.330 M= 20

	PHIXM	PHIZM0	PHIZM1	PHIZM2
-10	-0.64892E-04	0.22807E-04	0.13517E-04	-0.28321E-04
-9	-0.55712E-04	0.12977E-04	0.15917E-05	-0.49112E-04
-8	-0.40475E-04	-0.77779E-06	-0.14702E-04	-0.74822E-04
-7	-0.19946E-04	-0.19454E-04	-0.36323E-04	-0.10582E-03
-6	0.42184E-05	-0.44735E-04	-0.64688E-04	-0.14249E-03
-5	0.29777E-04	-0.79225E-04	-0.10174E-03	-0.18482E-03
-4	0.54287E-04	-0.12646E-03	-0.14982E-03	-0.23150E-03
-3	0.75375E-04	-0.19052E-03	-0.21123E-03	-0.27874E-03
-2	0.90839E-04	-0.27524E-03	-0.28765E-03	-0.31974E-03
-1	0.98676E-04	-0.38297E-03	-0.37951E-03	-0.34630E-03
0	0.97263E-04	-0.51315E-03	-0.48547E-03	-0.35270E-03
1	0.85857E-04	-0.66133E-03	-0.60220E-03	-0.33981E-03
2	0.65291E-04	-0.81894E-03	-0.72458E-03	-0.31709E-03
3	0.38460E-04	-0.97462E-03	-0.84623E-03	-0.30158E-03
4	0.99236E-05	-0.11170E-02	-0.96049E-03	-0.30923E-03
5	-0.15650E-04	-0.12378E-02	-0.10617E-02	-0.34333E-03
6	-0.35440E-04	-0.13340E-02	-0.11464E-02	-0.39444E-03
7	-0.49124E-04	-0.14070E-02	-0.12140E-02	-0.44952E-03
8	-0.57658E-04	-0.14609E-02	-0.12660E-02	-0.49944E-03
9	-0.61693E-04	-0.15005E-02	-0.13055E-02	-0.54104E-03
10	-0.61263E-04	-0.15304E-02	-0.13358E-02	-0.57531E-03

Here IX values of ±10 represent distances of one half the tunnel height. They lie just ahead of the fuselage nose and just behind the fuselage base. It can be seen that the axial values and gradients of the disturbance pressure are negligible and the upwash angles are small, the largest being near the base of approximately 0.1 degrees.

The values that would be incurred without flexing the walls are presented in the following table:

Table 3

NX= 40 NZ= 20 MACH=0.770 A=0.330 D=0.330 M= 20

	PHIXM	PHIZM0	PHIZM1	PHIZM2
-10	0.20055E-02	-0.32051E-03	-0.32980E-03	-0.37164E-03
-9	0.33860E-02	-0.16585E-03	-0.17724E-03	-0.22795E-03
-8	0.48950E-02	0.68470E-04	0.54544E-04	-0.55792E-05
-7	0.65023E-02	0.39917E-03	0.38230E-03	0.31280E-03
-6	0.81652E-02	0.84016E-03	0.82021E-03	0.74240E-03
-5	0.98299E-02	0.14003E-02	0.13778E-02	0.12947E-02
-4	0.11435E-01	0.20817E-02	0.20583E-02	0.19766E-02
-3	0.12915E-01	0.28776E-02	0.28569E-02	0.27894E-02
-2	0.14204E-01	0.37719E-02	0.37595E-02	0.37274E-02
-1	0.15244E-01	0.47391E-02	0.47426E-02	0.47758E-02
0	0.15989E-01	0.57455E-02	0.57731E-02	0.59059E-02
1	0.16413E-01	0.67526E-02	0.68117E-02	0.70741E-02
2	0.16517E-01	0.77220E-02	0.78164E-02	0.82239E-02
3	0.16328E-01	0.86205E-02	0.87489E-02	0.92935E-02
4	0.15891E-01	0.94239E-02	0.95804E-02	0.10232E-01
5	0.15263E-01	0.10119E-01	0.10295E-01	0.11013E-01
6	0.14500E-01	0.10703E-01	0.10890E-01	0.11642E-01
7	0.13656E-01	0.11178E-01	0.11372E-01	0.12136E-01
8	0.12775E-01	0.11554E-01	0.11749E-01	0.12516E-01
9	0.11897E-01	0.11840E-01	0.12035E-01	0.12799E-01
10	0.11054E-01	0.12044E-01	0.12239E-01	0.12999E-01

These disturbances would clearly alter the flow over the model to an extent that would make test data only marginally correctable.

The effect of using a finer panel grid can be seen by comparison of the first table of results with the following table computed with a total of 7260 panel pairs:

Table 4

NX= 60 NZ= 30 MACH=0.770 A=0.330 D=0.330 M= 20

	PHIXM	PHIZM0	PHIZM1	PHIZM2
-15	-0.41970E-04	-0.98716E-04	-0.10551E-03	-0.13814E-03
-14	-0.37653E-04	-0.11470E-03	-0.12213E-03	-0.15837E-03
-13	-0.32167E-04	-0.13305E-03	-0.14097E-03	-0.18022E-03
-12	-0.25633E-04	-0.15408E-03	-0.16223E-03	-0.20350E-03
-11	-0.18248E-04	-0.17815E-03	-0.18615E-03	-0.22797E-03
-10	-0.10275E-04	-0.20568E-03	-0.21296E-03	-0.25325E-03
-9	-0.20334E-05	-0.23715E-03	-0.24292E-03	-0.27886E-03
-8	0.61165E-05	-0.27306E-03	-0.27630E-03	-0.30413E-03
-7	0.13792E-04	-0.31395E-03	-0.31330E-03	-0.32817E-03
-6	0.20614E-04	-0.36030E-03	-0.35410E-03	-0.34982E-03
-5	0.26217E-04	-0.41253E-03	-0.39877E-03	-0.36771E-03
-4	0.30280E-04	-0.47088E-03	-0.44727E-03	-0.38027E-03
-3	0.32538E-04	-0.53537E-03	-0.49940E-03	-0.38598E-03
-2	0.32802E-04	-0.60568E-03	-0.55480E-03	-0.38359E-03
-1	0.30978E-04	-0.68114E-03	-0.61293E-03	-0.37249E-03
0	0.27074E-04	-0.76066E-03	-0.67309E-03	-0.35316E-03
1	0.21216E-04	-0.84275E-03	-0.73445E-03	-0.32740E-03
2	0.13655E-04	-0.92560E-03	-0.79608E-03	-0.29841E-03
3	0.47493E-05	-0.10072E-02	-0.85697E-03	-0.27047E-03
4	-0.50464E-05	-0.10854E-02	-0.91611E-03	-0.24813E-03
5	-0.15231E-04	-0.11584E-02	-0.97246E-03	-0.23523E-03
6	-0.25314E-04	-0.12244E-02	-0.10251E-02	-0.23399E-03
7	-0.34866E-04	-0.12822E-02	-0.10731E-02	-0.24455E-03
8	-0.43557E-04	-0.13313E-02	-0.11158E-02	-0.26522E-03
9	-0.51165E-04	-0.13713E-02	-0.11528E-02	-0.29317E-03
10	-0.57555E-04	-0.14028E-02	-0.11840E-02	-0.32524E-03
11	-0.62632E-04	-0.14264E-02	-0.12096E-02	-0.35869E-03
12	-0.66298E-04	-0.14433E-02	-0.12299E-02	-0.39153E-03
13	-0.68437E-04	-0.14548E-02	-0.12458E-02	-0.42255E-03
14	-0.68900E-04	-0.14620E-02	-0.12580E-02	-0.45121E-03
15	-0.67540E-04	-0.14665E-02	-0.12675E-02	-0.47738E-03

Only modest changes in the already small distortion values are evident.

3.2 Sensitivity to Span Load Shift

Naturally, the flow over the test model will not be exactly the same as that computed and used in setting the flex walls. The effect of any differences can be estimated by considering possible alterations in the local distributions for a given fixed lift coefficient. In the basic calculations the span of the wing trailing vortex pair was set to correspond to an elliptic span load distribution. To determine sensitivity to span load alteration the vortex pair spacing, set by the variable, SV, in AFMODL was reduced from π/4 times the wing span to 2/3 the wing span; however, the wall contours were held at the values set for the basic calculation. The resulting flow distortion for comparison with that of the basic case is presented in the following table:

Table 5

NX= 40 NZ= 20 MACH=0.770 A=0.330 D=0.330 M= 20

	PHIXM	PHIZM0	PHIZM1	PHIZM2
-10	-0.19797E-03	0.19535E-03	0.19034E-03	0.16217E-03
-9	-0.17327E-03	0.22410E-03	0.21795E-03	0.18447E-03
-8	-0.14063E-03	0.25440E-03	0.24677E-03	0.20811E-03
-7	-0.10121E-03	0.28463E-03	0.27513E-03	0.23196E-03
-6	-0.57348E-04	0.31170E-03	0.30007E-03	0.25387E-03
-5	-0.12184E-04	0.33057E-03	0.31702E-03	0.27072E-03
-4	0.30909E-04	0.33428E-03	0.31992E-03	0.27920E-03
-3	0.68808E-04	0.31449E-03	0.30186E-03	0.27712E-03
-2	0.98958E-04	0.26286E-03	0.25630E-03	0.26470E-03
-1	0.11951E-03	0.17313E-03	0.17863E-03	0.24434E-03
0	0.12944E-03	0.43387E-04	0.67660E-04	0.21832E-03
1	0.12876E-03	-0.12227E-03	-0.73454E-04	0.18599E-03
2	0.11903E-03	-0.31349E-03	-0.23731E-03	0.14193E-03
3	0.10359E-03	-0.51545E-03	-0.41344E-03	0.75798E-04
4	0.87118E-04	-0.71241E-03	-0.59019E-03	-0.20730E-04
5	0.74140E-04	-0.89178E-03	-0.75705E-03	-0.14418E-03
6	0.67138E-04	-0.10466E-02	-0.90657E-03	-0.28007E-03
7	0.66036E-04	-0.11756E-02	-0.10352E-02	-0.41225E-03
8	0.69496E-04	-0.12810E-02	-0.11427E-02	-0.53011E-03
9	0.76572E-04	-0.13665E-02	-0.12312E-02	-0.63020E-03
10	0.87040E-04	-0.14364E-02	-0.13038E-02	-0.71393E-03

Comparison of the two relevant Tables 2 and 5, shows differences of low order and within acceptable limits.

3.3 Sensitivity to Fore and Aft Load Shift

To test sensitivity to fore and aft shifting of the lift, the sweptline vortex was moved from the wing quarter chord to the wing half chord. The wall flexure remained at the setting for the basic flow calculation. The flow distortion under these conditions are presented in Table 6 for comparison with the basic case in Table 2.

Table 6

NX= 40 NZ= 20 MACH=0.770 A=0.330 D=0.330 M= 20

	PHIXM	PHIZM0	PHIZM1	PHIZM2
-10	-0.34504E-03	-0.89885E-04	-0.97977E-04	-0.13622E-03
-9	-0.38631E-03	-0.13871E-03	-0.14840E-03	-0.19393E-03
-8	-0.42445E-03	-0.20122E-03	-0.21285E-03	-0.26589E-03
-7	-0.45633E-03	-0.27938E-03	-0.29338E-03	-0.35379E-03
-6	-0.47747E-03	-0.37502E-03	-0.39170E-03	-0.45894E-03
-5	-0.48176E-03	-0.48939E-03	-0.50875E-03	-0.58144E-03
-4	-0.46157E-03	-0.62255E-03	-0.64386E-03	-0.71881E-03
-3	-0.40910E-03	-0.77266E-03	-0.79392E-03	-0.86419E-03
-2	-0.31875E-03	-0.93543E-03	-0.95285E-03	-0.10051E-02
-1	-0.19024E-03	-0.11040E-02	-0.11116E-02	-0.11242E-02
0	-0.31089E-04	-0.12692E-02	-0.12594E-02	-0.12025E-02
1	0.14291E-03	-0.14210E-02	-0.13856E-02	-0.12251E-02
2	0.31042E-03	-0.15496E-02	-0.14823E-02	-0.11892E-02
3	0.45004E-03	-0.16477E-02	-0.15459E-02	-0.11097E-02
4	0.54637E-03	-0.17118E-02	-0.15783E-02	-0.10141E-02
5	0.59356E-03	-0.17435E-02	-0.15851E-02	-0.92637E-03
6	0.59499E-03	-0.17493E-02	-0.15740E-02	-0.85758E-03
7	0.56030E-03	-0.17380E-02	-0.15530E-02	-0.80778E-03
8	0.50211E-03	-0.17182E-02	-0.15283E-02	-0.77220E-03
9	0.43324E-03	-0.16964E-02	-0.15047E-02	-0.74657E-03
10	0.36436E-03	-0.16769E-02	-0.14846E-02	-0.72886E-03

Once again the variations are seen to be minimal.

4.0 CONSIDERATION OF THREE DIMENSIONAL FLEXING OF THE UPPER AND LOWER WALLS

The two dimensional flexing previously considered in this report appears to lack the ability to prevent some residual aerodynamic wash-in (induced twist of the wings). It seemed therefore that three dimensional flexing might improve the situation. One practical scheme considered envisioned a single row of jacks along the flex wall centerline with the lateral edge of the wall plate held fixed but not clamped. The jacks would conform the plate at each axial jack location to satisfy the Tracor blockage algorithm. To provide input to the flow distortion program PHIXZM the plate slopes in the axial direction were required at every panel center. For this a program called JACK_DISPL was developed using classical thin plate theory. A description of the method used and a Fortran listing are presented in Appendix D.

Application of the analysis to the Langley 0.3-Meter TCT flexwall tunnel with the AEDC model at $C_L = 0.5$ indicated as expected the usual reduction in blockage and induced upwash; however, the spanwise variation of upwash was not improved at all even though it was basically small, e.g., only about 0.03 degrees in a tunnel quarter width away from the center. The output data are presented below in Tables 7 and 8.

Table 7 - Rigid Walls

MACH=0 CL=0.5

IX	PHIXM	PHIZM0	PHIZM1	PHIZM2
-10	0.193E-02	0.105E-02	0.107E-02	0.114E-02
-8	0.268E-02	0.147E-02	0.149E-02	0.158E-02
-6	0.344E-02	0.195E-02	0.198E-02	0.211E-02
-4	0.414E-02	0.248E-02	0.253E-02	0.270E-02
-2	0.473E-02	0.305E-02	0.311E-02	0.335E-02
0	0.513E-02	0.362E-02	0.370E-02	0.403E-02
2	0.531E-02	0.417E-02	0.427E-02	0.469E-02
4	0.527E-02	0.467E-02	0.479E-02	0.530E-02
6	0.504E-02	0.511E-02	0.524E-02	0.582E-02
8	0.467E-02	0.546E-02	0.562E-02	0.625E-02
10	0.424E-02	0.574E-02	0.591E-02	0.659E-02

Table 8 - Flexed Walls

MACH=0 CL=0.5

IX	PHIXM	PHIZX0	PHIZM1	PHIZM2
-10	0.532E-04	-0.738E-04	-0.946E-04	-0.172E-03
-8	-0.137E-03	-0.715E-04	-0.104E-03	-0.220E-03
-6	-0.274E-03	0.287E-04	-0.789E-04	-0.251E-03
-4	-0.329E-03	0.649E-04	-0.988E-05	-0.264E-03
-2	-0.285E-03	0.204E-03	0.978E-04	-0.264E-03
0	-0.133E-03	0.369E-03	0.225E-03	-0.261E-03
2	0.119E-03	0.536E-03	0.357E-03	-0.255E-03
4	0.436E-03	0.707E-03	0.495E-03	-0.234E-03
6	0.752E-03	0.892E-03	0.649E-03	-0.183E-03
8	0.991E-03	0.108E-02	0.815E-03	-0.102E-03
10	0.110E-02	0.126E-02	0.971E-03	-0.161E-04

It is probable that the lack of displacement of the plate edges reduces the effectiveness of the 3-D flexure.

5.0 ESTIMATION OF FLOW DISTORTION FROM TEST DATA

As mentioned in the introduction the concepts for using measured data on a suitable control surface during a test have been developed for some time. For application to hybrid conditions of partial adaptation the following analysis is presented to identify some functions used in the estimation of distortion that have the property of being calculable without iterative or inverting procedures that often introduce errors when applied to large matrices such as those needed for panel methods using large numbers of panels.

Consider a solid wall tunnel or a ventilated tunnel where wall pressures and corresponding normal velocity can be measured or determined during a test. It is assumed that nothing is known of the model flow field. Disturbance velocities, axial and normal to a control surface coincident with original walls are denoted ϕ_x and ϕ_n respectively and the following cases are defined:

ϕ_x^o and ϕ_n^o	are values at the wall for the wall-free condition (fully adapted).
ϕ_x^m and ϕ_n^m	are the values measured during the test at the control surface (partially adapted).
$\delta\phi_x^o$ and $\delta\phi_x^i$	are the outer and inner change in ϕ_x produced by the residual wall presence in its partially or non adapted state.
$\delta\phi_n$	is change in ϕ_n from partial to fully adapted state.
n	is defined positive inward.

Thus we may write

$$\delta\phi_x^i = \phi_x^m - \phi_x^o \tag{5}$$

and

$$\delta\phi_n = \phi_n^m - \phi_n^o \tag{6}$$

Our goal is the computation of ϕ_n or $\delta\phi_n$ for use in estimation of the residual wall interferences. The flow produced by the model in the region outside the control surface can also be considered to be produced by a distribution of normal velocity, ϕ_n over the control surface and we may write the matrix formula:

$$\phi_x^o(i) = \phi_n^o(j) \quad G^o(i,j) \tag{7}$$

where G^o is a Green's source function for the outer flow and i and j are field point and source location indices.

The change in the internal flow, $\delta\phi_x^i$, can be related to the normal velocity change, $\delta\phi_n$ as

$$\delta\phi_x^i(i) = \delta\phi_n(j) \; G^i(i,j). \tag{8}$$

where G^i is a Green's source function.

Introducing Equations (7) and (8) into (5) yields

$$\delta \; \phi_n(j) \; G^i(i,j) = -\phi_n^o \; (j) \; G^o(i,j) + \phi_x^m(i) \tag{9}$$

and using (6) to eliminate $\delta\phi_n$ we obtain

$$\phi_n^o(j) = (\phi_x^m(i) - \phi_n^m(j)\ G^i(i,j))[G^o(i,j) - G^i(i,j)]^{-1} \qquad (10)$$

This wall-free value is not exactly correct because the measurements are made with the model in a partially adapted state. It represents the wall-free flow about a model tested in a slightly non-uniform stream. However, the residual errors for practical cases can be very small.

The Green's function, G^i, can be generated directly by means of the method of images; however, the function, G^o, cannot. Fortunately as shown below the inverse function $[G^o - G^i]^{-1}$ can be generated directly.

The wall presence in its partially adapted state can also be represented by a distribution of doublets or vortex elements over the tunnel wall control surface. This representation produces a flow change that is discontinuous in ϕ_x and continuous in ϕ_n at the control surface and we may write:

$$\delta\phi_x^o(j) = \delta\phi_x^i(j) - \Gamma(j) \qquad (11)$$

where Γ is the local vortex intensity on the control surface. Also

$$\delta\phi_n^o = \delta\phi_n^i \equiv \delta\ \phi_n \qquad (12)$$

Outside the control surface Equation (7) applies also to the increment, $\delta\phi_x$ and

$$\delta\phi_x^o(j) = \delta\phi_n(i)\ G^o(i,j) \qquad (13)$$

Substitution of Equations (8) and (13) in (11) leads to

$$\Gamma(j) = \delta\phi_n(i)\ [G^i(i,j) - G^o(i,j)] \tag{14}$$

and by matrix inversion

$$\delta\ \phi_n(i) = \Gamma(j)\ [G^i(i,j) - G^o(i,j)]^{-1}, \tag{15}$$

but a relationship between Γ and $\delta\phi_n$ can be written as

$$\delta\phi_n(i) = -\Gamma(j)\ A\ (i,j) \tag{16}$$

where A is a function uniquely determined by the control surface geometry. It is indeed the normal velocity produced at a point on the wall, i, by a unit vortex element at j. It can be computed directly without images.

Comparison of Equations (15) and (16) yields

$$[G^i - G^o]^{-1} = -A \tag{17}$$

Finally from (10) and (17)

$$\phi_n^o = (\phi_x^m - \phi_n^m\ G^i)\ [A] \tag{18}$$

Equation (18) shows that the free air normal velocity distribution can be determined from the measurements and two directly calculable arrays, G^i and A. No iteration nor matrix inversion is required, thus a high degree of accuracy can be assured when using large numbers of elements. It is fast and straightforward to use elements or panels with sizes as small as one or two inches in an eight foot square wind tunnel.

To obtain the flow distortion at the position of the model, i.e., pressure and flow angle and their gradients, we may use the following expressions:

$$\phi_x = [\phi_n^o - \phi_n^m]\ [G^m] \tag{19}$$

$$\phi_n = (\phi_n^o - \phi_n^m)\ [H^m] \tag{20}$$

where G^m and H^m are Green's functions for wall sources. Both functions are developed by direct calculation using the method of images.

The complete code for computation of ϕ_n has been developed; it is called VEEXPHINO and a Fortran listing is presented in Appendix E. It requires the input functions measured wall normal velocity, MEASVN.DAT, and measured wall axial disturbance velocity, MEASVX.DAT. These functions must be developed to provide values at the centers of the panels chosen to correspond to the panel system chosen for VEEXPHINO. The output of VEEXPHINO called PHINWALL.DAT serves as input to PHIXZM for calculation of the residual flow distortion at the model location. If the system discussed is utilized in a practical case the pressures and normal velocities will probably be measured at a relatively small number of points. It is probably best to extrapolate the measured values to the more numerous chosen panel centers and then proceed rather than to limit the number of panels. The arrays G^i and A become very large if many panels are chosen, therefore, VEEXPHINO calculates smaller arrays from which G^i and A are generated by using the geometrical similarities of the panel arrangements in a tunnel with a constant cross-sectional shape.

The first part of VEEXPHINO computes the first term in parenthesis in Equation (18). It uses the same technique of images used for PHIXZM to derive the constituent array G0 used to generate the function $G^{i}(i,j)$. The second part of the code develops the matrix array A(i,j) by use of the equations for horseshoe vortices given in Reference 1. Each panel is assumed to contain a centrally located bound vortex and two trailing vortices that extend downstream to infinity. The equations are shown and described in some detail in the listing.

As mentioned previously, when the walls are not fully adapted the procedure described is only approximate in that the model produces a flow field influenced by the residual wall presence. However, the numerical experiments performed with theoretical models shows that partial adaptation in practical cases reduces the flow distortions to negligible values.

6.0 CONCLUSIONS

The results of the computations indicate the power of flexing upper and lower walls of a rectangular wind tunnel in reducing the flow distortions at a model test location. Axial pressure gradients can be reduced to negligible values and upwash and upwash gradients can be sharply reduced. Once the walls have been set for a given lift coefficient and using a good calculated approximation to the model flow field, the present results show that the residual flow distortions are insensitive to variations in span loading and fore and aft loading. Thus once set the data obtained can be quite accurate even though the flow about the model is somewhat different from that computed. The calculated residual flow distortion at the model offers a good measure of the quality of the data and when not too large can be used as a basis for corrections, i.e., small angle of attack corrections for the fuselage, an indication of the induced aerodynamic twist of the wings and correction for the induced tail angle of attack.

7.0 REFERENCES

1. Glauert, H., "The Elements of Airfoil and Airscrew Theory," Cambridge at the University Press, New York: The MacMillan Co., 1943.

2. Theordorsen, T., "The Theory of Wind-Tunnel Wall Interference," NACA Rep. No. 410, 1931.

3. Lock, C. N. H. and Beavan, J. A., "Tunnel Interference at Compressibility Speeds Using the Flexible Walls of the Rectangular High Speed Tunnel," British ARC R and M 2005, 1944.

4. Ferri, A. and Baronti, P.," A Method for Transonic Wind-Tunnel Corrections," AIAA Jour., Vol. 11, No. 1, 1973, pp. 63-66.

5. Sears, W. R., "Self-Correcting Wind Tunnels," The Sixteenth Lanchester Memorial Lecture of the Royal Aeronautical Society, London, May 1973. (Published as Aero. Jour., Vol. 78, Feb/Mar 1974, pp. 80-89.)

6. Capelier, G., Chevallier, J.-P., and Bouniol, F., "Nouvelle méthode de correction des effets de parois en courant plan," 14th Applied Aerodynamics Colloquium, Toulouse, Nov. 1977. (Available in English - ESA TT-491, European Space Agency, 1978, pp. 1-30).

7. Wedemyer, E. and Heddergoth, A., "Dehnbare Messstrecke fuer Windkanale, DFVLR IB 251-80 A01, February 1981.

8. Tuttle, M. H. and Mineck, R. E., "Adaptive Wall Wind Tunnels," A Selected Annotated Bibliography, NASA TM 87639, August 1986.

9. Harney, D. J., "Three Dimensional Test Experience with a Transonic Adaptive-Wall Wind Tunnel," AFWAL-TR-83-3028, March 1983.

10. Schairer, E. T. and Mendoza, J. P., "Adaptive-Wall Wind-Tunnel Research at Ames Research Center," in Wall Interference in Wind Tunnels, AGARD CP 335, 1982.

11. Wolf, S. W. D., Cook, I. D., and Goodyer, M. J., "The Status of Two- and Three-Dimensional Testing in the University of Southhampton Transonic Self-Streamlining Wind Tunnel," in Wall Interference in Wind Tunnels, AGARD CP 335, 1982.

12. Sickles, W. L., "A Data Base for Three-Dimensional Wall Interference Code Evaluation," in Wind Tunnel Wall Interference Assessment/Correction, 1983, NASA Conference Publication 2319, 1984, pp. 101-116.

13. Smith, J., "A Theoretical Exploration of the Capabilities of 2D Flexible Wall Test Sections for 3D Testing," NLR Rep. No. MP 84018U, 1984.

14. Timoshenko, S. P. and Woinowsky-Krieger, S., "Theory of Plates and Shells," 2nd Ed. McGraw Hill, New York, 1959, pp. 141-143.

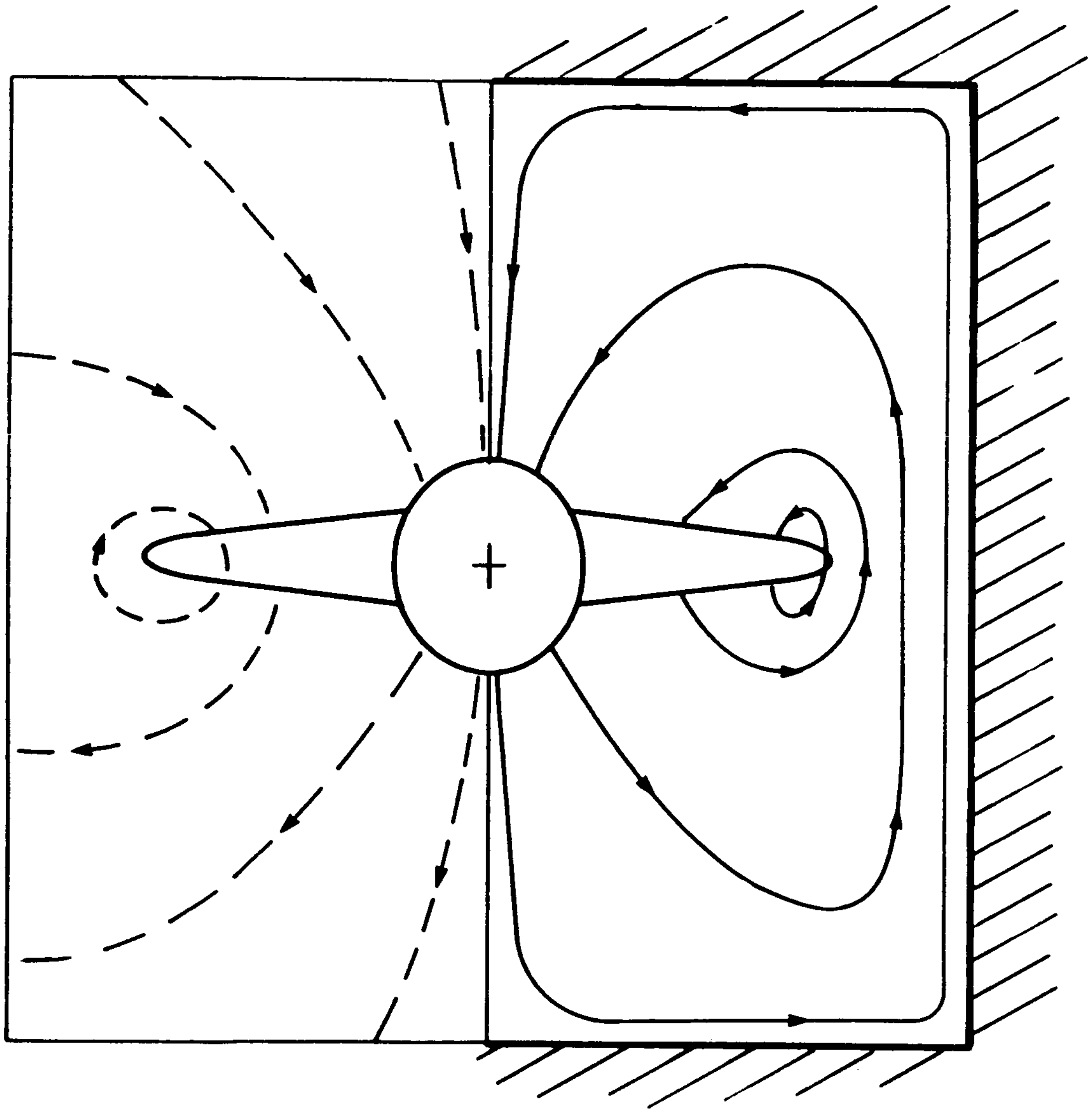


FIGURE 1 - STREAMLINE TRACES AT AN AXIAL STATION WITH WALL-FREE AND FIXED STRAIGHT WALL CONDITIONS

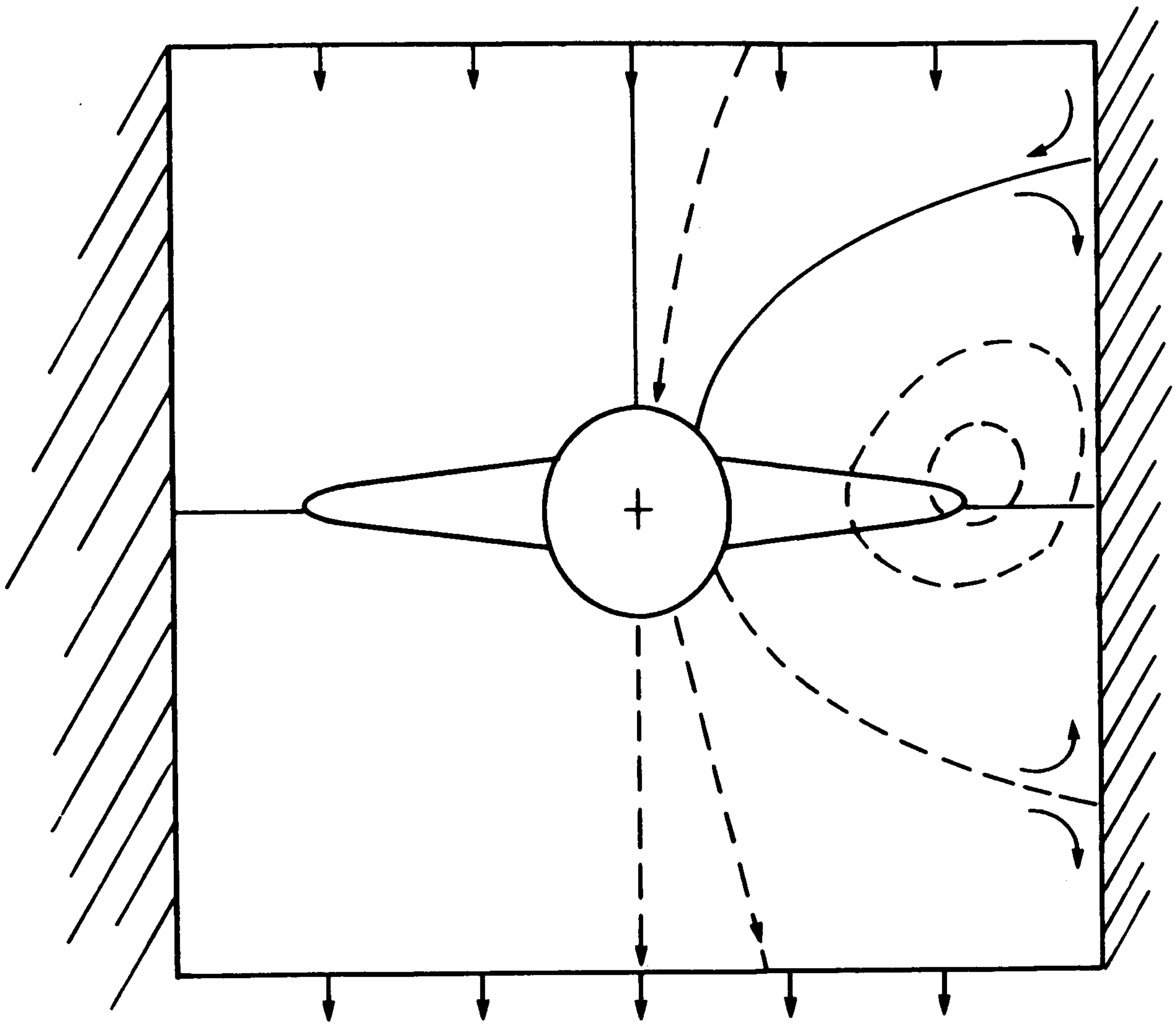


FIGURE 2 - STREAMLINE TRACES WITH MOVABLE UPPER AND LOWER WALLS

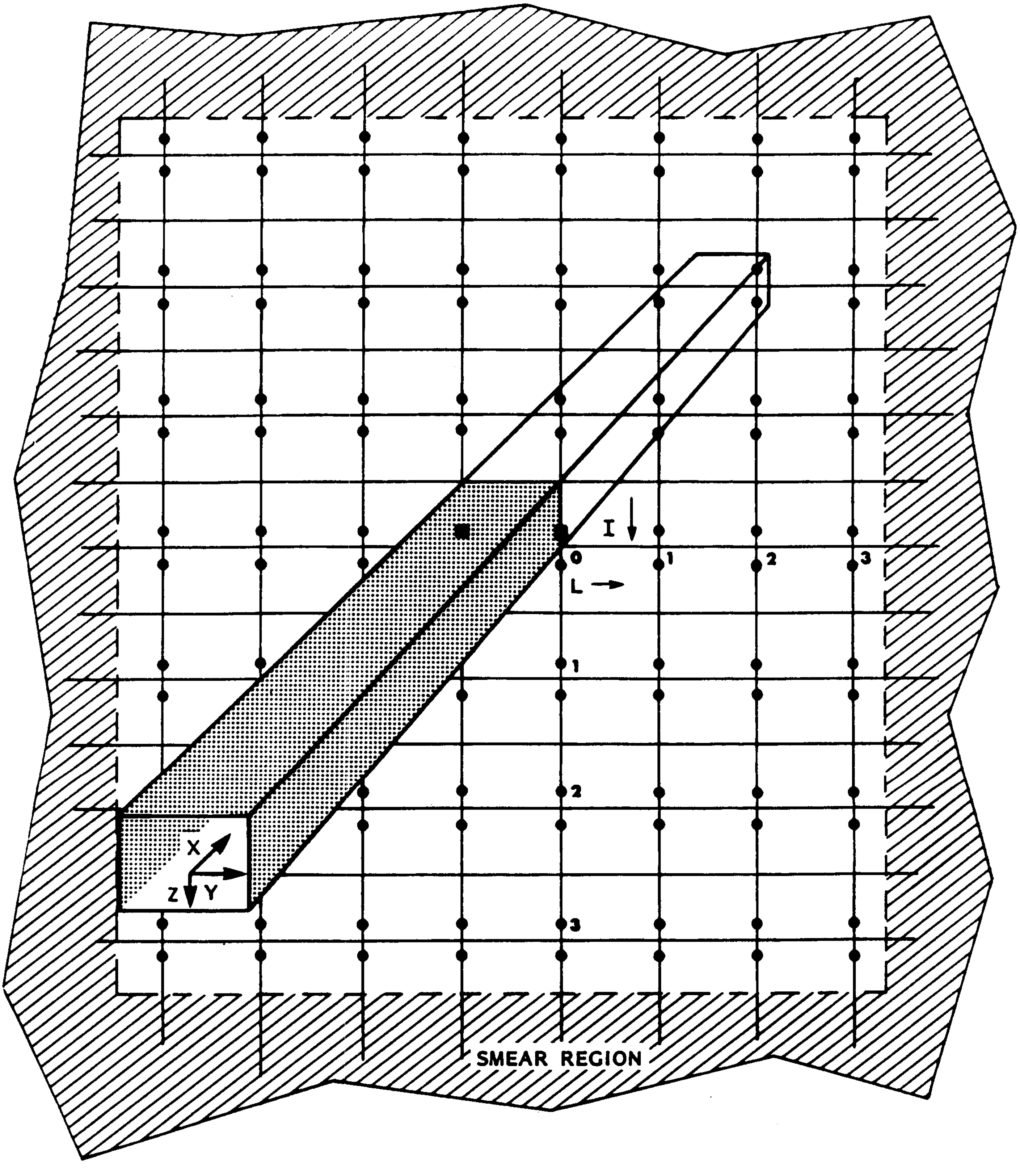


FIGURE 3 - SOURCE IMAGE SYSTEM FOR M = 3

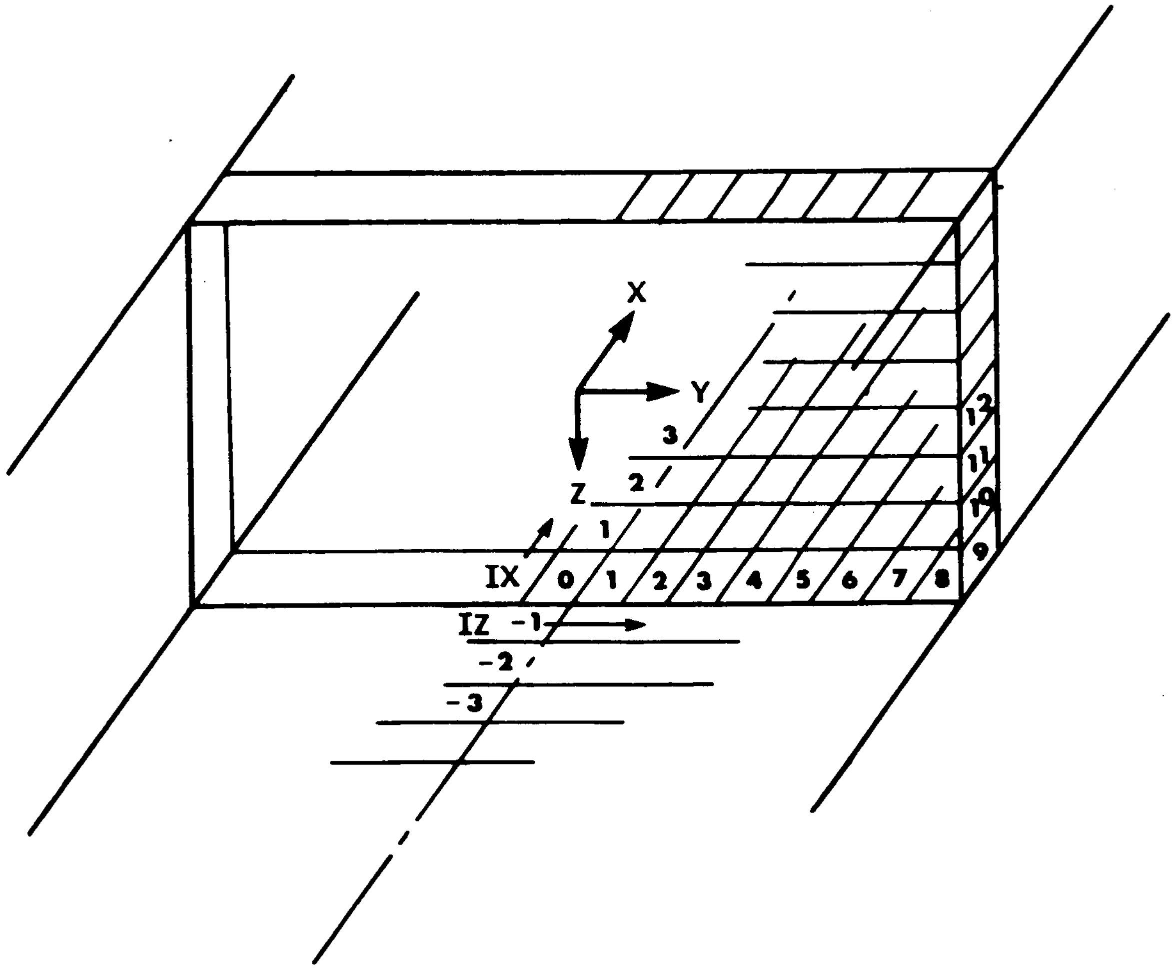
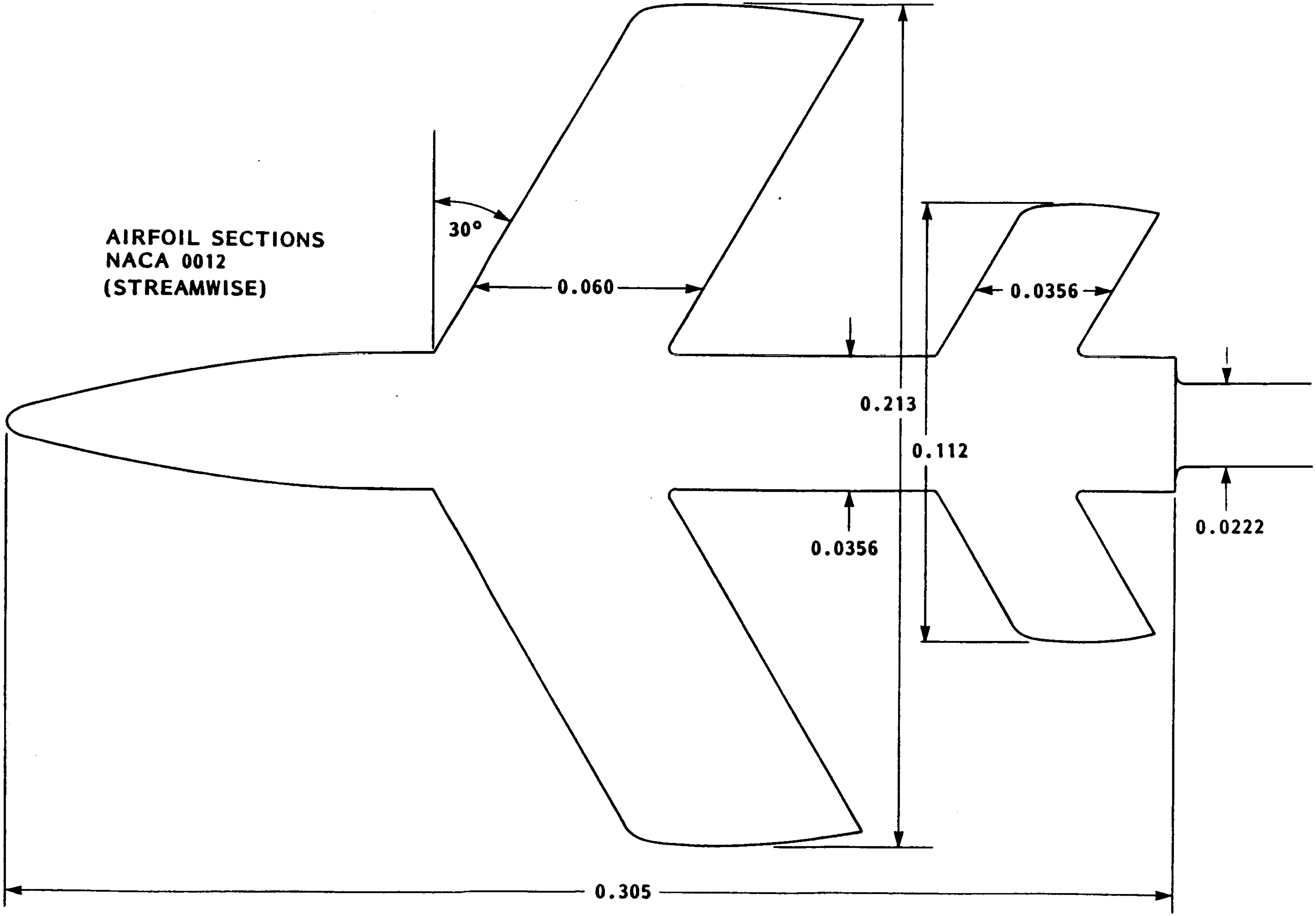


FIGURE 4 - COORDINATE AND PANEL NUMBERING SYSTEM



DIMENSIONS IN METERS

FIGURE 5 - AEDC MODEL AND DIMENSIONS

APPENDIX A - PHIXZM FORTRAN LISTING

Primary Symbols

A	Tunnel height
D	Tunnel breadth
G0	Axial disturbance velocities at $y + z = 0$ produced by a Green's source in the ring at IX = -NX
H0, H1, H2	Upwash velocities at $z = 0$, $y = 0$, $y = A/8$ and $y = A/4$ produced by the Green's source
MACH	Mach number
PHIXM	Tunnel centerline axial disturbance velocities produced by a distribution of normal velocities, VN, over all panels
PHIZM0	Centerline upwash due to the VN distribution
PHIZM1	Upwash at $z = 0$ and $y = A/8$
PHIZM2	Upwash at $z = 0$ and $y = A/4$
VN	Difference of calculated wall free velocity ratios and wall streamline slopes

```
C                        PROGRAM PHIXZM

C                 THIS PROGRAM COMPUTES THE AXIAL VELOCITY INCREMENT
C                 AND THE UPWASH IN THE MODEL VICINITY CAUSED BY WALLS
C                 PHIXM IS (+) FOR FLOWS DOWNSTREAM,AND PHIZM(UPWASH)
C                 IS (+) FOR INDUCED UPWARD FLOW AT THE MODEL.

C                 A RECTANGULAR TUNNEL IS ASSUMED OF HEIGHT=A AND
C                 WIDTH=D

C                 SELECT AN EVEN!!  NUMBER OF PANELS ON THE VERTICAL
C                 WALL(NZ).PANEL HEIGHT IS THUS A/NZ.PANEL LENGTH IS
C                 SET AT A/NZ ALSO. TUNNEL LENGTH IS SET BY NX, THE
C                 NUMBER OF PANELS FOREWARD AND AFT OF THE ORIGIN.


C                 PANEL SIZE ON THE WALL IS A/NZ BY A/NZ,SQUARE!
C                 ON THE FLOOR IT IS A/NZ BY D/NY

C                 PRANDTL-GLAUERT COMPRESSIBILTY CORRECTIONS ARE USED

C                 PROGRAM REQUIRES AN INPUT FUNCTION VN REPRESENTING
C                 THE NORMAL VELOCITIES THAT MUST BE CANCELED AT THE
C                 WALLS FOR ALL THE PANEL CENTER POSITIONS.THE VN ARRAY
C                 CAN BE CALCULATED FROM THE PROGRAM PHINO OR FROM DATA
C                 COMPUTED FROM AN AIRPLANE CODE LIKE THE BOPPE CODE OR
C                 OTHER SIMPLER REPRESENTATIONS.


                  DIMENSION VN(80,-80:80),X(-80:80), Y(80),Z(80),G0(40,-80:20),
          1         H0(40,-80:20),H1(40,-80:20),H2(40,-80:20),PT(-40:40,40),
          1         PJ(-40:40,40),Q(0:40,40),Q1(0:40,40),Q2(0:40,40),
          1         R(0:40,40),R1(0:40,40),R2(0:40,40),S1(0:40,40),
          1         S2(0:40,40),T1(0:40,40),T2(0:40,40), PHIXM(-20:20),
          1         PHIZM0(-20:20),PHIZM1(-20:20), PHIZM2(-20:20)

C                 NOTE! VALUES IN THE DIM STATEMENT ABOVE CORRESPOND TO
C                 A SQUARE TUNNEL WITH NZ=20, SOME '20' VALUES ARE THE LATERAL
C                 EXTENT OF IMAGES (M) TAKEN BEFORE SMEARING
C                 SOME MUST CHANGE WITH THE VALUE SELECTED
C                  FOR NZ.

                  PARAMETER (PI=3.14159)
                  REAL MACH
                  TYPE 10
10                FORMAT(10X,'ENTER NX NZ  MACH  A  D AND M')
                  READ(5,*) NX, NZ, MACH, A, D,M
                  BETA=SQRT(1-MACH*MACH)

                  NZ2=NZ/2.0
                  NY2=NINT(D*NZ2/A+0.1)
                  NY=NY2*2.00
                  DYDZ=D/A*NZ/NY

C                 NOW SET UP THE COORDINATES OF THE PANELS.

                  DO IX=-NX,NX
                    X(IX)=IX*A/NZ/BETA
                  END DO
                  DO IZ=1,NY2
```

```
          Y(IZ)=(IZ-0.5)*D/NY
          Z(IZ)=A/2
        END DO
        DO IZ=(NY2+1),(NY2+NZ)
          Y(IZ)=D/2
          Z(IZ)=(0.5*(NZ+NY+1)-IZ)*A/NZ
        END DO
        DO IZ=(NZ+NY2+1),(NZ+NY)
          Y(IZ)=((NZ+NY+0.5)-IZ)*D/NY
          Z(IZ)=-A/2
        END DO

C       SET UP SOME COMMON REPEATING TERMS TO SAVE TIME


        DO JZ=1,NZ2+NY2
          YD=Y(JZ)
          YD1=YD-A/8
          YD2=YD-A/4
          YP1=YD+A/8
          YP2=YD+A/4

        DO I=-M,M
          PT(I,JZ)=(2*A*I+Z(JZ))*(2*A*I+Z(JZ))
          PJ(I,JZ)=(2*A*I+A-Z(JZ))*(2*A*I+A-Z(JZ))
        END DO
        DO L=0,M
          DL=D*L
          DP=D*(L+1)
          Q(L,JZ)=(DL+YD)*(DL+YD)
          Q1(L,JZ)=(DL+YD1)*(DL+YD1)
          Q2(L,JZ)=(DL+YD2)*(DL+YD2)
          R(L,JZ)= (DP-YD)*(DP-YD)
          R1(L,JZ)=(DP-YP1)*(DP-YP1)
          R2(L,JZ)=(DP-YP2)*(DP-YP2)
          S1(L,JZ)=(DL+YP1)*(DL+YP1)
          S2(L,JZ)=(DL+YP2)*(DL+YP2)
          T1(L,JZ)=(DP-YD1)*(DP-YD1)
          T2(L,JZ)=(DP-YD2)*(DP-YD2)
        END DO
        END DO

C       THE PRIMARY COMPUTATION BEGINS HERE

        DO KX=-NX,NZ2
C       THESE CALCULATIONS BELOW GIVE VALUES FROM THE FIRST
C       RING OF SOURCE PANELS AT IX=-NX

         XX=X(-NX)-X(KX)
         XS=XX*XX

C       FIRST WE CALCULATE THE SMEARED SOURCES

          EE=A/PI/D/NZ/NZ/BETA
          DM=(M+1)*D
          HM=(M+0.75)*2*A
          HM2=(M+0.25)*2*A

        GS=EE/BETA*(ATAN(-XX*SQRT(XS+DM*DM+HM*HM)/(HM*DM))+
     1    ATAN(-XX*SQRT(XS+DM*DM+HM2*HM2)/(HM2*DM)))
```

```
C             GS IS THE PHIX VALUES FROM THE SMEARED SOURCE
C               IMAGES.

              G=(XS+HM2**2)
              F=DM+SQRT(DM**2+G)
              G2=(XS+HM**2)
              F2=DM+SQRT(DM**2+G2)
              HS=-EE*ALOG(F*SQRT(G2)/(F2*SQRT(G)))

C             HS IS THE PHIZ VALUES FROM THE SMEAR.

              DO JZ=1,NY2
              DO I=-M,M
                P1=PT(I,JZ)+XS
                P2=PJ(I,JZ)+XS
              DO L=0,M
                TT1=P1+Q(L,JZ)
                TT1=TT1*SQRT(TT1)
                TT2=P1+Q1(L,JZ)
                TT2=TT2*SQRT(TT2)
                TT3=P1+Q2(L,JZ)
                TT3=TT3*SQRT(TT3)
                TT4=P2+R(L,JZ)
                TT4=TT4*SQRT(TT4)
                TT5=P2+R1(L,JZ)
                TT5=TT5*SQRT(TT5)
                TT6=P2+R2(L,JZ)
                TT6=TT6*SQRT(TT6)
                TT8=P1+S1(L,JZ)
                TT8=TT8*SQRT(TT8)
                TT9=P1+S2(L,JZ)
                TT9=TT9*SQRT(TT9)
                TT11=P2+T1(L,JZ)
                TT11=TT11*SQRT(TT11)
                TT12=P2+T2(L,JZ)
                TT12=TT12*SQRT(TT12)

              QQ=A*A/(2*PI*BETA*NZ*NZ)
              TTT=(2/TT1+2/TT4)*QQ
              RR=(1/TT2+1/TT5+1/TT8+1/TT11)*QQ
              SS=(1/TT3+1/TT6+1/TT9+1/TT12)*QQ

              GO(JZ,KX)=-XX*TTT/BETA+GO(JZ,KX)
              HO(JZ,KX)=A*(2*I+0.5)*TTT+HO(JZ,KX)
              H1(JZ,KX)=A*(2*I+0.5)*RR+H1(JZ,KX)
              H2(JZ,KX)=A*(2*I+0.5)*SS+H2(JZ,KX)

              END DO
              END DO

C             THE TERM DYDZ APPEARS BELOW TO ACCOUNT FOR  THE
C             DIFFERENCE IN PANEL WIDTH IF ANY OF FLOOR AND WALL PANELS


              GO(JZ,KX)=(GO(JZ,KX)+GS)*DYDZ
              HO(JZ,KX)=(HO(JZ,KX)+HS)*DYDZ
              H1(JZ,KX)=(H1(JZ,KX)+HS)*DYDZ
              H2(JZ,KX)=(H2(JZ,KX)+HS)*DYDZ
              END DO
```

```
                DO JZ=NY2+1,NY2+NZ2
                DO I=-M,M
                  P1=PT(I,JZ)+XS
                  P2=PJ(I,JZ)+XS
                DO L=0,M
                  TT1=P1+Q(L,JZ)
                  TT1=TT1*SQRT(TT1)
                  TT2=P1+Q1(L,JZ)
                  TT2=TT2*SQRT(TT2)
                  TT3=P1+Q2(L,JZ)
                  TT3=TT3*SQRT(TT3)
                  TT4=P2+R(L,JZ)
                  TT4=TT4*SQRT(TT4)
                  TT5=P2+R1(L,JZ)
                  TT5=TT5*SQRT(TT5)
                  TT6=P2+R2(L,JZ)
                  TT6=TT6*SQRT(TT6)
                  TT8=P1+S1(L,JZ)
                  TT8=TT8*SQRT(TT8)
                  TT9=P1+S2(L,JZ)
                  TT9=TT9*SQRT(TT9)
                  TT11=P2+T1(L,JZ)
                  TT11=TT11*SQRT(TT11)
                  TT12=P2+T2(L,JZ)
                  TT12=TT12*SQRT(TT12)

                TTT=(2/TT1+2/TT4)*QQ
                RR1=(1/TT2+1/TT8)*QQ
                RR2=(1/TT5+1/TT11)*QQ
                SS1=(1/TT3+1/TT9)*QQ
                SS2=(1/TT6+1/TT12)*QQ

                G0(JZ,KX)=-XX*TTT/BETA+G0(JZ,KX)
                H0(JZ,KX)=QQ*((2*A*I+Z(JZ))*2/TT1+(2*A*(I+.5)-Z(JZ))*2/TT4)+
        1          H0(JZ,KX)
                H1(JZ,KX)=(2*A*I+Z(JZ))*RR1+(2*A*(I+.5)-Z(JZ))*RR2+H1(JZ,KX)
                H2(JZ,KX)=(2*A*I+Z(JZ))*SS1+(2*A*(I+.5)-Z(JZ))*SS2+H2(JZ,KX)

                END DO
                END DO

                G0(JZ,KX)=G0(JZ,KX)+GS
                H0(JZ,KX)=H0(JZ,KX)+HS
                H1(JZ,KX)=H1(JZ,KX)+HS
                H2(JZ,KX)=H2(JZ,KX)+HS

                END DO
                END DO

C               THIS COMPLETES THE CALCULATION FOR SOURCES LOCATED
C               AT JX=-NX IN  THE LOWER HALF OF THE TUNNEL

                OPEN(UNIT=2,NAME='PHINWALL.DAT',STATUS='OLD')
                READ(2,*) ((VN(I,J), I=1,NY+NZ) , J=-NX,NX)
                CLOSE (2)


                DO IX=-NZ2,NZ2
                DO JX=-NX,IX
```

```
                      KX=-NX-(JX-IX)
                    DO JZ=1,(NY2+NZ2)
                     PHIXM(IX)=PHIXM(IX)-VN(JZ,JX)*G0(JZ,KX)
                     PHIZM0(IX)=PHIZM0(IX)-VN(JZ,JX)*H0(JZ,KX)
                     PHIZM1(IX)=PHIZM1(IX)-VN(JZ,JX)*H1(JZ,KX)
                     PHIZM2(IX)=PHIZM2(IX)-VN(JZ,JX)*H2(JZ,KX)
                    END DO
                    DO JZ=(NY2+NZ2+1),(NZ+NY)
                      KZ=NY+NZ+1-JZ
                      PHIXM(IX)=PHIXM(IX)-VN(JZ,JX)*G0(KZ,KX)
                      PHIZM0(IX)=PHIZM0(IX)+VN(JZ,JX)*H0(KZ,KX)
                      PHIZM1(IX)=PHIZM1(IX)+VN(JZ,JX)*H1(KZ,KX)
                      PHIZM2(IX)=PHIZM2(IX)+VN(JZ,JX)*H2(KZ,KX)
                    END DO
                    END DO
                    DO JX=(IX+1),NX
                      LX=-NX+JX-IX
                    DO JZ=1,(NY2+NZ2)
                     PHIXM(IX)=PHIXM(IX)+VN(JZ,JX)*G0(JZ,LX)
                     PHIZM0(IX)=PHIZM0(IX)-VN(JZ,JX)*H0(JZ,LX)
                     PHIZM1(IX)=PHIZM1(IX)-VN(JZ,JX)*H1(JZ,LX)
                     PHIZM2(IX)=PHIZM2(IX)-VN(JZ,JX)*H2(JZ,LX)
                    END DO
                    DO JZ=(NY+NZ+2)/2,(NZ+NY)
                      KZ=NY+NZ+1-JZ
                      PHIXM(IX)=PHIXM(IX)+VN(JZ,JX)*G0(KZ,LX)
                      PHIZM0(IX)=PHIZM0(IX)+VN(JZ,JX)*H0(KZ,LX)
                      PHIZM1(IX)=PHIZM1(IX)+VN(JZ,JX)*H1(KZ,LX)
                      PHIZM2(IX)=PHIZM2(IX)+VN(JZ,JX)*H2(KZ,LX)
                    END DO
                    END DO
                    END DO

                    OPEN(UNIT=1,NAME='PHIXZM.DAT',STATUS='NEW')
                    WRITE(1,47) (CR, NX, NZ, MACH, A, D, M)
47                  FORMAT(5X,'CR=',F5.3,'NX=',I3,'NZ=',I3,'MACH=',F5.3,
        1           'A=',F5.3,'D=',F5.3,'M=',I3)
                    WRITE(1,50)
50                  FORMAT(10X'PHIXM',10X,'PHIZM0',10X,'PHIZM1'10X,'PHIZM2')
                    WRITE(1,60) (IX,PHIXM(IX),PHIZM0(IX),PHIZM1(IX),PHIZM2(IX),
        1                   IX=-NZ/2,NZ/2)
60                  FORMAT(7X,I5,4E15.5)
                    CLOSE(UNIT=1)

                    END
```

APPENDIX B - AFMODL FORTRAN LISTING

Primary Symbols

A	Tunnel height and breadth (assumed equal here)
AR & ART	Wing and tail aspect ratios
C & CT	Wing and tail chords
CIRC & CIRCT	Circulation of wing and tail vortices
CL	Lift coefficient
DX	Extension of panels downstream of NX
DZXL & DZXU	X-wise slope of flexwalls
MACH	Mach number
S, ST	Wing and tail spans
SV, SVT	Wing and tail vortex spans (spacing)
VN	Computed wall-free normal velocities at panel centers due to test model
VNA	Antisymmetrical component of VN
VNS	Symmetrical component of VN
VNN	Difference between wall-free normal velocities and wall-slopes due to flexing or boundary layer growth
WT, WTT	Wing and tail air foil thicknesses
XJ	Location of Langley 0.3-meter TCT flexwall jacks

```
                   PROGRAM AFMODL

C            COMPUTES THE WALL- FREE VELOCITIES(VN) NORMAL TO THE
C            WALLS OF A SQUARE WIND TUNNEL OF AN AEDC MODEL
C            TO BE USED IN A LANGLEY EXPERIMENTAL TEST.
C            IT ALSO COMPUTES THE RESIDUAL NORMAL VELOCITIES AFTER
C            APPLICATION OF THE TRACOR ALGORITHMFOR BLOCKAGE(VNN)

C            IT PRINTS OUT A FILE CALLED VNZERO.DAT, THE WALL-FREE
C            NORMAL VELOCITIES, FOR COMPARISON WITH THE VALUES FROM
C            A NONLINEAR CODE SUCH AS 'TUNCOR'.

C            THIS PROGRAM USES 2*NX+1 PLUS DX PANELS OVER THE LENGTH OF
C            THE WIND TUNNEL AND NY+NZ PANELS AROUND THE HALF
C            PERIMETER.  TO INCREASE THE NUMBER OF PANELS
C            NEW VALUES MUST BE PUT INTO THE DIMENSION STATEMENT
C            AND THE LIMITS NX, DX AND NZ .

C            THIS PROGRAM REPRESENTS THE MODEL BY TWO SOURCES(BODY)
C            VEE LINE-VORTICES(WING&TAIL LIFT) AND +&- LINE SOURCES
C            (WING&TAIL THICKNESS).BODY LIFT BY LINE DOUBLETS
C            IT ASSUMES THE CP IS LOCATED AT THE ZERO POSITION OF THE
C             X COORDINATES

C            THE WING IS ASSUMED TO CARRY 80% OF THE TOTAL LIFT

             DIMENSION X(-80:120),XJ(-20:0),Y(80),Z(80),VNS(80,-80:120),
     1         VNA(80,-80:120),VN(80,-80:120),DZXL(-80:120),DZXU(-80:120),
     1         ZL(-80:120),ZU(-80:120),VNN(80,-80:120)

C            XJ(IX) ARE THE 1/3 METER TUNNEL JACK LOCATIONS
C            X(IX),Y(IZ) AND Z(IZ) ARE THE CENTERS OF THE PANELS

C            THE TRACOR ALGORITHM IS APPLIED IN THIS PROGRAM
C            THE NUMBERS BELOW ARE FIXED BY THE DIMENSIONS OF THE
C            AEDC MODEL

             PARAMETER (PI=3.14159,S=0.10668, SV=.0837863,AR=3.5,
     1         D=0.03556,C=.060198,CT=0.03556,WT=.0072238,WTT=.004267,
     1        ST=.05588,SVT=.043888,A=.33, ART=3.1429)
             REAL  MACH
             TYPE 10
10           FORMAT(10X,'ENTER  CL  MACH  NX   NZ')

             READ(5,*)  CL, MACH, NX, NZ
             BETA=SQRT(1-MACH*MACH)

C            THE FOLLOWING RESULT FROM THE MODEL DIMENSIONS

             XRVC=-.03759     !LOCATION OF VORTEX &C/L INTERSECTION(METERS)
             XRV=XRVC/BETA
             XRVTC=0.0889  !SAME FOR THE TAIL
             XRVT=XRVTC/BETA
             XSL=-.10795        !LOCATION OF FOREWARD SOURCE
             XST=0.1524             ! METERS, SAME FOR THE REAR SOURCE
             TG=.57735/BETA  ! TANGENT OF THE STRETCHED SWEEPANGLE
C                             REAL SWEEP IS 30 DEGREES!

             TS=1+TG*TG
```

```
      DX=0
      NZ2=NZ/2
      NY=NZ
      NY2=NY/2

C     NOW SET UP THE COORDINATES OF THE POINTS AT WHICH
C       VELOCITIES ARE TO BE CALCULATED.

      DO IZ=1,NZ2
        Z(IZ)=A/2
        Y(IZ)=A/NZ*(IZ-0.5)
      END DO
      DO IZ=NZ2+1,NY2+NZ
        Y(IZ)=A/2
        Z(IZ)=A/NZ*(NZ+0.5-IZ)
      END DO
      DO IZ=NZ+NY2+1,NZ+NY
        Z(IZ)=-A/2
        Y(IZ)=(NZ+NY+0.5-IZ)*A/NZ
      END DO
      DO IX=-NX,NX+DX
        X(IX)=A/NZ*IX   ! FOR PANEL CENTERS
      END DO
      XJ(-20)=-27.75*.0254
      XJ(-19)=-22*.0254
      XJ(-18)=-17*.0254
      XJ(-17)=-13*.0254
      XJ(-16)=-10*.0254
      XJ(-15)=-8*.0254
      XJ(-14)=-6.5*.0254
      XJ(-13)=--5*.0254
      XJ(-12)=-3.5*.0254
      XJ(-11)=-2*.0254
      XJ(-10)=-.5*.0254
      XJ(-9)=1.0*.0254
      XJ(-8)=3*.0254
      XJ(-7)=5*.0254
      XJ(-6)=7*.0254
      XJ(-5)=10*.0254
      XJ(-4)=14*.0254
      XJ(-3)=19*.0254
      XJ(-2)=24*.0254
      XJ(-1)=29*.0254
      XJ(0)=34.5*.0254    ! ALL FOR JACK LOCATIONS

      CIRC=0.8*CL*S*S/(SV*AR)
      ALPHA=0.8*CL/(3.2+1.755*MACH**6)   !IN RADIANS
C     THIS ASSUMES AN APPROXIMATE MACH DEPENCENCE FOR CL/ALPHA

      Q=WT
      QT=WTT
      CIRCT=.2*CL*ST*ST/(SVT*ART)  !TAIL CIRCULATION
      DDD=1+.5*D*D*BETA*BETA/((XST-XSL)*(XST-XSL))  !APPROX STRENGTH CORRECTOR

      DO IX=-NX,NX+DX

C     STRETCHED DISTANCES FROM FIELD POINT TO INTERSECTION OF C/L AND:
        XV=(XRVC-X(IX))/BETA        ! WING VORTEX
        XL=(XRVC-C/4-X(IX))/BETA    ! WING LEADING EDGE
        XT=(XRVC+3*C/4-X(IX))/BETA  ! WING TRAILING EDGE
```

```
            XLS=(XSL-X(IX))/BETA         ! BODY SOURCE
            XTS=(XST-X(IX))/BETA         ! BODY SINK(AFT)
            XVT=(XRVTC-X(IX))/BETA       ! TAIL VORTEX
            XLT=XVT-CT/4/BETA            ! TAIL LEADING EDGE
            XTT=XVT+3*CT/4/BETA          ! TAIL TRAILING EDGE

C          RM IS THE ARM LENGTH FROM THE POINT OF ROTATION OF
C          THE BALANCE/STING SYSTEM TO THE CP ORIGIN OF COORDS.
C          THE MODEL IS DISPLACED UPWARDS BY A DISTANCE ALPHA*RM

           RM=0.501269 !METERS

           DO IZ=1,NY2
            F=(XV+Y(IZ)*TG)
            F1=(XV-Y(IZ)*TG)
            FT=(XVT+Y(IZ)*TG)
            FT1=(XVT-Y(IZ)*TG)
            YY=Y(IZ)**2
            ZZ=(Z(IZ)+ALPHA*RM)**2
            Q1=4*(F*F+TS*ZZ)
            Q2=4*(F1*F1+TS*ZZ)
            Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
            Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
            Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
            Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
            Q1T=4*(FT*FT+TS*ZZ)
            Q2T=4*(F1T*F1T+TS*ZZ)
            Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
            Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
            Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
            Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
            R1=SQRT(XV**2+ZZ+YY)
            R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
            R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
            R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
            R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
            R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
            R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
            R7=ZZ+YY+XLS*XLS
            R8=ZZ+YY+XTS*XTS

            R1T=SQRT(XVT**2+ZZ+YY)
            R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
            R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
            R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
            R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
            R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
            R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)


           VNS(IZ,IX)=-D*D*DDD*(Z(IZ)+ALPHA*RM)/16/BETA*(1/(R7*SQRT(R7))
     1      -0.61/(R8*SQRT(R8)))
```

```
C                THESE ABOVE ARE THE BODY SOURCE TERMS
                 VNS(IZ,IX)=-Q*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*S+XL*TG-Y(IZ))/(Q3*R4)-
          1        ((TS*D/2+XL*TG-Y(IZ))/(Q3*R3)+(TS*S+XL*TG+Y(IZ))/(Q4*R41)-
          1        (TS*D/2+XL*TG+Y(IZ))/(Q4*R31)-(TS*S+XT*TG-Y(IZ))/(Q5*R6)+
          1        (TS*D/2+XT*TG-Y(IZ))/(Q5*R5)-(TS*S+XT*TG+Y(IZ))/(Q6*R61)+
          1        (TS*D/2+XT*TG+Y(IZ))/(Q6*R51))+VNS(IZ,IX)
C                THESE WERE THE WING LINE SOURCE TERMS

                 VNS(IZ,IX)=-QT*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*ST+XLT*TG-Y(IZ))/
          1        (Q3T*R4T)-(TS*D/2+XLT*TG-Y(IZ))/(Q3T*R3T)+(TS*ST+XLT*TG+Y(IZ))/
          1        (Q4T*R41T)-(TS*D/2+XLT*TG+Y(IZ))/(Q4T*R31T)-(TS*ST*XTT*TG-
          1        Y(IZ))/(Q5T*R6T)+(TS*D/2+XTT*TG-Y(IZ))/(Q5T*R5T)-
          1        (TS*ST+XTT*TG+Y(IZ))/(Q6T*R61T)+(TS*D/2+XTT*TG+Y(IZ))/
          1        (Q6T*R51T))+VNS(IZ,IX)
C                THESE WERE THE TAIL LINE SOURCE TERMS

                 VNA(IZ,IX)=D*D*ALPHA/8*((YY-ZZ)/(YY+ZZ)**2*(1-XLS/SQRT(R7))
          1       +ZZ*XLS/(YY+ZZ)/(R7*SQRT(R7))-0.61*((YY-ZZ)/(YY+ZZ)**2*
          1       (1-XTS/SQRT(R8))+ZZ*XTS/(YY+ZZ)/(R8*SQRT(R8))))
C                THESE ARE THE DOUBLET(BODY LIFT) TERMS


                 TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
                 TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

                 VNA(IZ,IX)=VNA(IZ,IX)+CIRC/4/PI*(F*TT+F1*TO+(Y(IZ)-SV)/
          1        (ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)-(Y(IZ)+SV)/(ZZ+
          1         (Y(IZ)+SV)**2)*(1-(XV+SV*TG)/R21))
C                THE ABOVE ARE THE WING LINE VORTEX TERMS

                 TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
                 TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

                 VNA(IZ,IX)=VNA(IZ,IX)+CIRCT/4/PI*(FT*TTT+F1T*TOT+(Y(IZ)-SVT)/
          1        (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)-(Y(IZ)+SVT)/
          1        (ZZ+(Y(IZ)+SVT)**2)*(1-(XVT+SVT*TG)/R21T))
C                THE ABOVE ARE THE TAIL VORTEX TERMS

                 VN(IZ,IX)=VNA(IZ,IX)+VNS(IZ,IX)

                 END DO

                 DO IZ=NY2+NZ+1,NY+NZ   !UPPER WALL
                   F=(XV+Y(IZ)*TG)
                   F1=(XV-Y(IZ)*TG)
                   FT=(XVT+Y(IZ)*TG)
                   FT1=(XVT-Y(IZ)*TG)
                   YY=Y(IZ)**2
                   ZZ=(Z(IZ)+ALPHA*RM)**2
                   Q1=4*(F*F+TS*ZZ)
                   Q2=4*(F1*F1+TS*ZZ)
                   Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
                   Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
                   Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
                   Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
                   Q1T=4*(FT*FT+TS*ZZ)
                   Q2T=4*(F1T*F1T+TS*ZZ)
                   Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
                   Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
                   Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
```

```
      Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
      R1=SQRT(XV**2+ZZ+YY)
      R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
      R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
      R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
      R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
      R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
      R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
      R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
      R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
      R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
      R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
      R7=ZZ+YY+XLS*XLS
      R8=ZZ+YY+XTS*XTS

      R1T=SQRT(XVT**2+ZZ+YY)
      R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
      R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
      R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
      R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
      R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
      R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
      R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
      R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
      R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
      R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)


      VNS(IZ,IX)=+D*D*DDD*(Z(IZ)+ALPHA*RM)/16/BETA*(1/(R7*SQRT(R7))
     1  -0.61/(R8*SQRT(R8)))
C     THESE ABOVE ARE THE BODY SOURCE TERMS
      VNS(IZ,IX)=Q*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*S+XL*TG-Y(IZ))/(Q3*R4)-
     1  (TS*D/2+XL*TG-Y(IZ))/(Q3*R3)+(TS*S+XL*TG+Y(IZ))/(Q4*R41)-
     1  (TS*D/2+XL*TG+Y(IZ))/(Q4*R31)-(TS*S+XT*TG-Y(IZ))/(Q5*R6)+
     1  (TS*D/2+XT*TG-Y(IZ))/(Q5*R5)-(TS*S+XT*TG+Y(IZ))/(Q6*R61)+
     1  (TS*D/2+XT*TG+Y(IZ))/(Q6*R51))+VNS(IZ,IX)
C     THESE WERE THE WING LINE SOURCE TERMS

      VNS(IZ,IX)=QT*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*ST+XLT*TG-Y(IZ))/
     1  (Q3T*R4T)-(TS*D/2+XLT*TG-Y(IZ))/(Q3T*R3T)+(TS*ST+XLT*TG+Y(IZ))/
     1  (Q4T*R41T)-(TS*D/2+XLT*TG+Y(IZ))/(Q4T*R31T)-(TS*ST*XTT*TG-
     1  Y(IZ))/(Q5T*R6T)+(TS*D/2+XTT*TG-Y(IZ))/(Q5T*R5T)-
     1  (TS*ST+XTT*TG+Y(IZ))/(Q6T*R61T)+(TS*D/2+XTT*TG+Y(IZ))/
     1  (Q6T*R51T))+VNS(IZ,IX)
C     THESE WERE THE TAIL LINE SOURCE TERMS

      VNA(IZ,IX)=-D*D*ALPHA/8*((YY-ZZ)/(YY+ZZ)**2*(1-XLS/SQRT(R7))
     1 +ZZ*XLS/(YY+ZZ)/(R7*SQRT(R7))-0.61*((YY-ZZ)/(YY+ZZ)**2*
     1 (1-XTS/SQRT(R8))+ZZ*XTS/(YY+ZZ)/(R8*SQRT(R8))))
C     THESE ARE THE DOUBLET(BODY LIFT) TERMS


      TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
      TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

      VNA(IZ,IX)=VNA(IZ,IX)-CIRC/4/PI*(F*TT+F1*TO+(Y(IZ)-SV)/
     1  (ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)-(Y(IZ)+SV)/(ZZ+
     1   (Y(IZ)+SV)**2)*(1-(XV+SV*TG)/R21))
C     THE ABOVE ARE THE WING LINE VORTEX TERMS
```

```
      TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
      TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

      VNA(IZ,IX)=VNA(IZ,IX)-CIRCT/4/PI*(FT*TTT+F1T*TOT+(Y(IZ)-SVT)/
     1  (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)-(Y(IZ)+SVT)/
     1  (ZZ+(Y(IZ)+SVT)**2)*(1-(XVT+SVT*TG)/R21T))
C     THE ABOVE ARE THE TAIL VORTEX TERMS

      VN(IZ,IX)=+VNA(IZ,IX)+VNS(IZ,IX)

      END DO

      DO IZ=NY2+1,NY2+NZ    !COMPLETE SIDEWALL
        YY=Y(IZ)**2
        ZZ=(Z(IZ)+ALPHA*RM)**2
        F=(XV+Y(IZ)*TG)
        F1=(XV-Y(IZ)*TG)
        FT=(XVT+Y(IZ)*TG)
        FT1=(XVT-Y(IZ)*TG)

        Q1=4*(F*F+TS*ZZ)
        Q2=4*(F1*F1+TS*ZZ)
        Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
        Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
        Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
        Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
        Q1T=4*(FT*FT+TS*ZZ)
        Q2T=4*(F1T*F1T+TS*ZZ)
        Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
        Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
        Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
        Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
        R1=SQRT(XV**2+ZZ+Y(IZ)**2)
        R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
        R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
        R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
        R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
        R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
        R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
        R7=ZZ+YY+XLS*XLS
        R8=ZZ+YY+XTS*XTS

        R1T=SQRT(XVT**2+ZZ+YY)
        R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
        R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
        R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
        R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
        R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
        R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)

      VNS(IZ,IX)=-D*D*DDD/16/BETA*Y(IZ)*(1/(R7*SQRT(R7))-
     1  0.61/(R8*SQRT(R8)))
```

```
C           THESE ARE THE BODY SOURCE TERMS ON THE WALL.

            VNS(IZ,IX)=VNS(IZ,IX)-Q/PI/BETA*(((XL+Y(IZ)*TG)*(XL+S*TG)+ZZ)/
     1        Q3/R4-((XL+Y(IZ)*TG)*(XL+D/2*TG)+ZZ)/Q3/R3-((XL-Y(IZ)*TG)*
     1        (XL+S*TG)+ZZ)/Q4/R41+((XL-Y(IZ)*TG)*(XL+D/2*TG)+ZZ)/
     1        Q4/R31-((XT+Y(IZ)*TG)*(XT+S*TG)+ZZ)/Q5/R6+((XT+Y(IZ)*TG)*
     1        (XT+D/2*TG)+ZZ)/Q5/R5+((XT-Y(IZ)*TG)*(XT+S*TG)+ZZ)/Q6/R61
     1        -((XT-Y(IZ)*TG)*(XT+D/2*TG)+ZZ)/Q6/R51)
C           THE ABOVE ARE THE WING THICKNESS (LINE SOURCE) TERMS

            VNS(IZ,IX)=VNS(IZ,IX)-QT/PI/BETA*(((XLT+Y(IZ)*TG)*(XLT+ST*TG)
     1        +ZZ)/Q3T/R4T-((XLT+Y(IZ)*TG)*(XLT+D/2*TG)+ZZ)/Q3T/R3T-
     1        ((XLT-Y(IZ)*TG)*(XLT+ST*TG)+ZZ)/Q4T/R41T+((XLT-Y(IZ)*TG)*
     1        (XLT+D/2*TG)+ZZ)/Q4T/R31T-((XTT+Y(IZ)*TG)*(XTT+ST*TG)+ZZ)/
     1        Q5T/R6T+((XTT+Y(IZ)*TG)*(XTT+D/2*TG)+ZZ)/Q5T/R5T+((XTT-
     1        Y(IZ)*TG)*(XTT+ST*TG)+ZZ)/Q6T/R61T-((XTT-Y(IZ)*TG)*(XTT+
     1        D/2*TG)+ZZ)/Q6T/R51T)
C           THESE ARE THE TAIL THICKNESS TERMS

            VNA(IZ,IX)=-D*D*ALPHA/8*Y(IZ)*(Z(IZ)+ALPHA*RM)/(YY+ZZ)*((1-XLS/
     1       SQRT(R7))*2/(YY+ZZ)-XLS/R7/SQRT(R7)-0.61*((1-XTS/SQRT(R8))*2/
     1       (YY+ZZ)-XTS/R8/SQRT(R8)))
C           THE ABOVE ARE THE DOUBLET TERMS(BODY LIFT)

            TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
            TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

            TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
            TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

            VNA(IZ,IX)=VNA(IZ,IX)+CIRC/4/PI*(Z(IZ)+ALPHA*RM)*(TG*(TT-TO)
     1       -1/(ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)+1/(ZZ+(Y(IZ)+SV)**2)
     1       *(1-(XV+SV*TG)/R21))
C           THESE ARE THE TERMS FROM THE SWEPT LINE VORTEX

            VNA(IZ,IX)=+CIRCT/4/PI*(Z(IZ)+ALPHA*RM)*(TG*(TTT-TOT)-1/
     1        (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)+1/(ZZ+(Y(IZ)+SVT)
     1        **2)*(1-(XVT+SVT*TG)/R21T))+VNA(IZ,IX)
C           THESE ARE FOR THE TAIL VORTEX SYSTEM

            VN(IZ,IX)=VNA(IZ,IX)+VNS(IZ,IX)

            END DO               !COMPLETE TUNNEL IS NOW DONE

            END DO

C           WE NOW APPLY THE TRACOR ALGORITHM AND COMPUTE THE
C           SHAPE OF THE WALL TO ILLIMINATE BLOCKAGE. DZXL AND DZXU
C           ARE THE SLOPES OF THE LOWER AND UPPER WALLS.

            DO IX=-NX,NX
            DO IZ=1,NY2+NZ2
              DZXL(IX)=DZXL(IX)-VN(IZ,IX)/NY2
            END DO

C           THIS INTEGRATES THE NORMAL VELOCITIES OVER THE LOWER HALF
C           OF THE TUNNEL AND DIVIDES BY THE LOWER WALL WIDTH

            DO IZ=NY2+NZ2+1,NY+NZ
              DZXU(IX)=DZXU(IX)+VN(IZ,IX)/NY2
```

```
       END DO

       DO IZ=1,NY2
         VNN(IZ,IX)=+VN(IZ,IX)+DZXL(IX)
       END DO

       DO IZ=NY2+1,NY2+NZ
         VNN(IZ,IX)=+VN(IZ,IX)
       END DO

       DO IZ=NY2+NZ+1,NY+NZ
         VNN(IZ,IX)=+VN(IZ,IX)-DZXU(IX)
       END DO
       END DO

C      THIS CALCULATES THE RESIDUAL NORMAL VELOCITIES ON THE TUNNEL
C      WALL THAT MUST BE NEGATED BY THE GREEN'S SOURCES


C      WE NOW USE SIMPSONS RULE TO GET THE DISPLACEMENTS OF THE WALLS

       ZL(-NX)=0
       ZL(-NX+1)=(DZXL(-NX)+DZXL(-NX+1))*A/NZ/2
       ZU(-NX)=0
       ZU(-NX+1)=(DZXU(-NX)+DZXU(-NX+1))*A/NZ/2
       DO IX=-NX+2,NX
         ZL(IX)=ZL(IX-2)+A/3/NZ*(DZXL(IX-2)+4*DZXL(IX-1)+
     1      DZXL(IX))
         ZU(IX)=ZU(IX-2)+A/3/NZ*(DZXU(IX-2)+4*DZXU(IX-1)+
     1      DZXU(IX))
       END DO

       OPEN(UNIT=7,NAME='ZDISPL.DAT',STATUS='NEW')
       WRITE(7,200)
200    FORMAT(10X,'.33 METER TUNNEL AND AIRFORCE MODEL')
       WRITE(7,210) CL,  MACH
210    FORMAT(20X,'CL=',F4.2,5X,'MACH='F4.2)

       WRITE(7,220)
220    FORMAT(10X,'I',8X,'ZL(I)',8X,'ZU(I)')
       WRITE(7,230) (I,ZL(I),ZU(I),I=-NX,NX)
230    FORMAT(7X,I5,2E15.5)
       CLOSE(7)

       OPEN(UNIT=1,NAME='VNZERO.DAT',STATUS='NEW')
       WRITE(1,240)
240    FORMAT(10X,'WALL FREE NORMAL VELOCITIES,LOWER HALF')
       WRITE(1,245) CL,   MACH, NX,  NZ
245    FORMAT(30X,'CL=',F4.2,5X,'MACH=',F4.2,5X,'NX='I4,5X,'NZ='I4)

       DO IX=-NX,NX
       WRITE(1,250) IX,(VN(IZ,IX),IZ=1,NY2+NZ2)
250    FORMAT(2X,I3,2X,9E13.4,/,(7X,9E13.4))
       END DO

       WRITE(1,255)
255    FORMAT(10X,'WALLFREE NORMAL VELOCITIES,UPPER HALF')

       DO IX=-NX,NX
       WRITE(1,250) IX,(VN(IZ,IX),IZ=(NY2+NZ2+1),(NY+NZ))
```

```
END DO
CLOSE(1)


OPEN(UNIT=2,NAME='FHINWALL.DAT',STATUS='NEW')
WRITE(2,*) ((VNN(IZ,IX),IZ=1,(NY+NZ)),IX=-NX,NX)
CLOSE(2)

END
```

APPENDIX C - NONLVN FORTRAN LISTING

Primary Symbols

A	Tunnel height and breadth (assumed equal here)
AR & ART	Wing and tail aspect ratios
C & CT	Wing and tail chords
CIRC & CIRCT	Circulation of wing and tail vortices
CL	Lift coefficient
DX	Extension of panels downstream of NX
DZXL & DZXU	X-wise slope of flex walls
MACH	Mach number
S, ST	Wing and tail spans
SV, SVT	Wing and tail vortex spans (spacing)
VN	Computed wall-free normal velocities at panel centers due to test model
VNA	Antisymmetrical component of VN
VNS	Symmetrical component of VN
VNN	Difference between wall-free normal velocities and wall-slopes due to flexing or boundary layer growth
WT, WTT	Wing and tail airfoil thicknesses
XJ	Location of Langley 0.3-meter TCT flexwall jacks
VNZERO	Wall-free normal velocities at panel centers from nonlinear flow computation

```
              PROGRAM NONLVN

C             THIS PROGRAM COMPUTES THE WALL SHAPES FOR THE 0.3 METER
C             TUNNEL WITH THE USE OF A NONLINEAR CODE SUCH AS TUNCOR
C             THAT PROVIDIES THE WALL-FREE NORMAL VELOCITIES IN A FILE
C             CALLED VNZERONL.DAT.  THE VALUES ARE COMPUTED FOR
C             A MODEL SHAPE AND LIFT COEFFICIENT. COMPARISONS
C             OF RESULTS WITH TEST DATA SHOULD BE MADE AT THE
C             SAME LIFT COEFFICIENT!

C             THIS PROGRAM USES 2*NX+1 PLUS DX PANELS OVER THE LENGTH OF
C             THE WIND TUNNEL AND NY+NZ PANELS AROUND THE HALF
C             PERIMETER.  TO INCREASE THE NUMBER OF PANELS
C             NEW VALUES MAY BE NEEDED IN THE DIMENSION STATEMENT
C             AND THE LIMITS NX, DX AND NZ .

              DIMENSION X(-80:120),Y(80),Z(80),VNZERO(80,-80:80),
     1          DZXL(-80:120),DZXU(-80:120),
     1          ZL(-80:120),ZU(-80:120),VNN(80,-80:120)

              PARAMETER(PI=3.14159)

              REAL  MACH
              TYPE 10
10            FORMAT(10X,'ENTER  CL  MACH  NX   NZ')
              READ(5,*)  CL, MACH, NX, NZ
              BETA=SQRT(1-MACH*MACH)



              DX=0
              NZ2=NZ/2
              NY=NZ
              NY2=NY/2

C             NOW SET UP THE COORDINATES OF THE POINTS AT WHICH
C               VELOCITIES ARE TO BE CALCULATED.

              DO IZ=1,NZ2
                Z(IZ)=A/2
                Y(IZ)=A/NZ*(IZ-0.5)
              END DO
              DO IZ=NZ2+1,NY2+NZ
                Y(IZ)=A/2
                Z(IZ)=A/NZ*(NZ+0.5-IZ)
              END DO
              DO IZ=NZ+NY2+1,NZ+NY
                Z(IZ)=-A/2
                Y(IZ)=(NZ+NY+0.5-IZ)*A/NZ
              END DO
              DO IX=-NX,NX+DX
                X(IX)=A/NZ*IX    ! ALL DIMENSIONS FOR PANEL CENTERS
              END DO

              OPEN(UNIT=1,NAME='VNZERONL.DAT',STATUS='OLD')
              READ(1,*) ((VNZERO(IZ,IX),IZ=1,NY+NZ),IX=-NX,NX)
              CLOSE(1)

C             WE NOW APPLY THE TRACOR ALGORITHM AND COMPUTE THE
```

```
C     SHAPE OF THE WALL TO ILLIMINATE BLOCKAGE. DZXL AND DZXU
C     ARE THE SLOPES OF THE LOWER AND UPPER WALLS.

      DO IX=-NX,NX
      DO IZ=1,NY2+NZ2
        DZXL(IX)=DZXL(IX)-VNZERO(IZ,IX)/NY2
      END DO

C     THIS INTEGRATES THE NORMAL VELOCITIES OVER THE LOWER HALF
C     OF THE TUNNEL AND DIVIDES BY THE LOWER WALL WIDTH

      DO IZ=NY2+NZ2+1,NY+NZ                     ! UPPER HALF
        DZXU(IX)=DZXU(IX)+VNZERO(IZ,IX)/NY2
      END DO

      DO IZ=1,NY2
        VNN(IZ,IX)=+VNZERO(IZ,IX)+DZXL(IX)
      END DO

      DO IZ=NY2+1,NY2+NZ
        VNN(IZ,IX)=+VNZERO(IZ,IX)
      END DO

      DO IZ=NY2+NZ+1,NY+NZ
        VNN(IZ,IX)=+VNZERO(IZ,IX)-DZXU(IX)
      END DO
      END DO

C     THIS CALCULATES THE RESIDUAL NORMAL VELOCITIES ON THE TUNNEL
C     WALL THAT MUST BE NEGATED BY THE GREEN'S SOURCES


C     WE NOW USE SIMPSONS RULE TO GET THE DISPLACEMENTS OF THE WALLS

      ZL(-NX)=0
      ZL(-NX+1)=(DZXL(-NX)+DZXL(-NX+1))*A/NZ2
      ZU(-NX)=0
      ZU(-NX+1)=(DZXU(-NX)+DZXU(-NX+1))*A/NZ2
      DO IX=-NX+2,NX
        ZL(IX)=ZL(IX-2)+A/3/NZ*(DZXL(IX-2)+4*DZXL(IX-1)+
     1    DZXL(IX))
        ZU(IX)=ZU(IX-2)+A/3/NZ*(DZXU(IX-2)+4*DZXU(IX-1)+
     1    DZXU(IX))
      END DO

      OPEN(UNIT=7,NAME='ZDISPL.DAT',STATUS='NEW')
      WRITE(7,200)
200   FORMAT(10X,'.33 METER TUNNEL AND AIRFORCE MODEL')
      WRITE(7,210) CL,  MACH
210   FORMAT(20X,'CL=',F4.2,5X,'MACH='F4.2)

      WRITE(7,220)
220   FORMAT(10X,'I',8X,'ZL(I)',8X,'ZU(I)')
      WRITE(7,230) (I,ZL(I),ZU(I),I=-NX,NX)
230   FORMAT(7X,I5,2E15.5)
      CLOSE(7)


      OPEN(UNIT=2,NAME='PHINWALL.DAT',STATUS='NEW')
      WRITE(2,*) ((VNN(IZ,IX),IZ=1,NY+NZ)),IX=-NX,NX)
```

```
CLOSE(2)
END
```

APPENDIX D - JACK_DISPL ANALYSIS AND FORTRAN LISTING

Determination of Wind Tunnel Wall Displacements

In this Appendix, we outline the analysis upon which the wall displacement program JACK_DISPL is based. The two primary outputs of this program are the displacements of both the floor and ceiling jacks needed to relieve the blockage and an estimated residual interference velocity normal to the floor and ceiling. This is then used as an input to the routine PHIXZM to estimate residual flow distortion at the model.

The starting point is the basic relation adopted for relating the streamwise slope of the wind tunnel flexible wall (here floor and ceiling) to the wall-free normal velocities induced by the model. Thus for unit free stream velocity

$$-\int_{S/2} V_n \, d\ell = \int_{-b/2}^{b/2} \frac{\partial w}{\partial x} \, dy \qquad (D.1)$$

where V_n = Inflow velocity due to model normalized on free stream velocity.

w = Wall displacement from flat initial position (taken to be positive outward); w is a function of x and y.

$d\ell$ = Differential length along the perimeter S of the tunnel cross section at each streamwise station x.

x = Streamwise position along wall; x = 0 is taken here as being located at the beginning of the flexible wall.

y = Spanwise coordinate; y = 0 is located at center of flexible wall of total span b.

The integration on the left hand side of (D.1) is performed over the half perimeter symbolically denoted S/2. When considering deflection of the floor, the S/2 is taken as the lower half of the tunnel; S/2 is taken over the upper half of the tunnel to determine flexure of the ceiling. This is the application of the Tracor Blockage Algorithm to the problem.

We define a spanwise integrated displacement as

$$w^*(x) = \int_{-b/2}^{b/2} w(x,y)\,dy = 2\int_{o}^{b/2} w(x,y)\,dy \qquad (D.2)$$

and the integrated normal velocity as

$$V^*(x) = \int_{S/2} V_n\,d\ell\ . \qquad (D.3)$$

combining (D.1 - D.3) and rearranging,

$$V^* = -\frac{\partial}{\partial x}\int_{-b/2}^{b/2} w\,dy = -\frac{\partial w^*}{\partial x} \qquad (D.4)$$

$$w^*(x) = -\int_{o}^{x} V^*(x')\,dx' \qquad (D.5)$$

Thus the spanwise integrated wall displacements have been expressed in terms of the input wall-free velocities normal to the tunnel walls.

We now relate these values of w^* to adjustments that are obtainable with a given single streamwise series of jacks that are located on the floor and ceiling at $y = 0$. To do this, we will model each flexible wall as a simply supported rectangular plate subjected to a concentrated ("point") load at each jack location. Inspection of (D.5) shows that it is not the load, but the displacements that are important. Thus we seek the displacement at each jack location, $w(y = 0)$, that enables the resulting plate shape to satisfy (D.5). Loads will be employed only as intermediate variables used to obtain displacements. The actual loads needed for a given displacement depend strongly on the details of the plate construction - stiffness, thickness, ribbing, etc. No attempt is made to calculate actual loads as they are not needed.

The solution for the displacement of a simply supported rectangular plate of length a and width b subjected to a point load P at $(x,y) = (\zeta,0)$ is given in Reference 14 for $y \geq 0$ as

$$w(x,y;\zeta) = \frac{Pa^2}{2\pi^3 D} \sum_{m=1}^{\infty} \left\{ \left[(1+\alpha_m \tanh \alpha_m) \sinh \left(\frac{\alpha_m}{b} (b-2y) \right) - \frac{\alpha_m}{b} (b-2y) \cosh \left(\frac{\alpha_m}{b} (b-2y) \right) \right] \frac{\sin \left(\frac{m\pi\zeta}{a} \right) \sin \left(\frac{m\pi x}{a} \right)}{m^3 \cosh \alpha_m} \right\}$$

$$\alpha_m = m \frac{\pi b}{2a} \quad . \qquad \text{(D.6)}$$

Here, D is the plate flexural rigidity. We can define a dimensionless displacement due to a load of unit force as

$$\hat{w}(x,y;\zeta) = \frac{D}{Pa^2} w(x,y;\zeta) = \frac{1}{2\pi^3} \sum_{m=1}^{\infty} \{\} \qquad \text{(D.7)}$$

where the summation expression is the same as in (D.6).

Since the differential equation whose solution is given by (D.6) is linear, we can superpose solutions. Thus the displacement due to a series of loads $P_i = P(x = \zeta_i)$ can be expressed

$$w(x,y) = \sum_i \bar{P}_i \cdot \hat{w}(x,y;\zeta_i) \qquad \text{(D.8)}$$

where

$$\tilde{P}_i = \frac{P_i \, a^2}{D} \quad , \qquad \text{(D.9)}$$

a normalized load that has units of length.

If w is evaluated at a finite number of discrete points (x_j, y_k), relation (D.8) represents a set of linear algebraic equations,

$$w(x_j, y_k) = \sum_i \hat{w}\,(x_j, y_k; \zeta_i) \cdot \tilde{P}_i \quad , \qquad \text{(D.10)}$$

or in matrix notation,

$$\underline{w} = \underline{\bar{\hat{w}}} \cdot \underline{\tilde{P}} \qquad \text{(D.11)}$$

$\underline{w}$ is a column vector whose elements are arranged in the order $(x_1,y_1,)$, $(x_1,y_2),\ldots(x_1,y_{N_k}),(x_2,y_1),\ldots,(x_j,y_k),\ldots,(x_{N_j},y_{N_k})$.

Its length is $N_j \cdot N_k$ where N_j and N_k are the total number of discrete x and y locations, respectively, of interest. $\underline{P}$ is a column vector of length N_i, the total number of concentrated loads. $\underline{w}$ is a rectangular matrix of length $N_j \cdot N_k$ and width N_i. It should be noted that in general, the locations $\{x_j\}$ and $\{\zeta_i\}$ need not be the same.

If the integration of (D.2) is applied to (D.6), another matrix equation is similarly found

$$\underline{w^*} = \underline{\bar{\hat{w}}^*} \cdot \underline{\tilde{P}} \qquad \text{(D.12)}$$

where $\underline{P}$ is as previously defined, $\underline{w}^*$ is of length N_j, and $\bar{\hat{\underline{w}}}^*$ is of size N_j by N_i. The "*" denotes spanwise integrated quantities analogous to those of (D.11).

The slope of the flexible walls, $\partial w/\partial x\ (x,y)$, is found by differentiating (D.10) leading to an analog of (D.11):

$$\underline{w}_x = \bar{\hat{\underline{w}}}_x \quad \bar{\underline{P}} \tag{D.13}$$

where the elements of $\underline{w}_x$ are $\partial w/\partial x$. The expressions for the elements of the matrices $\hat{\underline{w}}^*$ and $\hat{\underline{w}}_x$ have been obtained by analytically integrating or differentiating the elements of $\bar{\hat{\underline{w}}}$ given by (D.6,D.7).

The algorithm implemented in JACK_DISPL and its subroutines is now outlined. Calculations are repeated for the floor and ceiling.

1. Using as input the normal velocities in free air at the wall locations, epxression (D.3) is calculated at each jack location by a Simpson's integration.

2. w^* is found at each jack location using a trapezoidal integration of (D.5).

3. The elements of the various matrices in (D.11 - D.13) are calculated using expressions (D.6,D.7) and the appropriate integrated and differentiated forms of (D.6,D.7) defined above. The infinite series is truncated at, typically, 150 terms. Numerical experimentation has shown this to be well converged. Simplified expressions for the hyperbolic functions are used where appropriate for large arguments.

4. $\underset{\sim}{\tilde{P}}$ is obtained from (D.12) using $\underline{w}^*$ from (D.5) and IMSL Subroutine "LEQT2F".

5. The needed displacements at the jack locations are determined from (D.11) with $(x_j, y_k) = (x_{jack}, 0)$. Displacements at other locations can be similarly evaluated.

6. The "residual" or adjusted normal velocities are calculated using (D.13), and the expression (based on the original normal velocities and the effect of the sloping walls):

$$V_{n,residual}\ (x,y) = V_{n,original}\ (x,y) + \frac{\partial w}{\partial x}\ (x,y). \qquad \text{(D.14)}$$

In general, the signs of the two right hand terms will be opposite such that the new "residual" V_n will be much less than the original.

Fortran Listings

Listings for the programs JACK_DISPL and AFMODLJ follow. AFMODLJ is a modified version of AFMODL that provides the input wall-free normal velocities at the panel and jack locations of JACK_DISPL.

```
               PROGRAM  JACK_DISPL

C   THIS PROGRAM READS THE DATA FILE [VNJ.DAT] FOR THE LOCATION OF THE JACKS
C   IN THE WIND TUNNEL AND THE MAGNITUDES OF THE NORMAL VELOCITIES TO BE USED
C   FOR CALCULATING THE JACK DISPLACEMENT NEEDED TO RELIEVE THE EXCESS OUTWARD
C   FLOW IN THE TUNNEL.
C
C   THE DATA FILE [VNO.DAT] IS THEN READ FOR THE DIMENSIONS OF THE WIND
C   TUNNEL, THE NUMBER OF PANELS IN THE X, Y, & Z  DIRECTIONS AND THE
C   LOCATIONS OF THE CENTER OF THE PANELS IN THE X & Y DIRECTION, THE WIDTHS
C   OF THE PANELS IN THE Y & Z DIRECTIONS, AND THE MAGNITUDE OF THE NORMAL
C   VELOCITIES AT THE PANEL CENTERS ON THE FLOOR AND CEILING.
C
C   THE PROGRAM THEN CALLS THE SUBROUTINE 'SIMPSON' TO INTEGRATE THE NORMAL
C   VELOCITIES FOR THE LOWER HALF AND UPPER HALF OF THE TUNNEL AT EACH JACK
C   LOCATION USING SIMPSON'S INTEGRATION METHOD.  THE SUBROUTINE 'MATRIX' IS
C   THEN CALLED TO CALCULATE THE MATRICES FOR THE UNIT POINT LOAD DISPLACEMENT,
C   THE UNIT POINT LOAD DISPLACEMENT DIFFERENTIATED WITH RESPECT TO X, AND THE
C   UNIT POINT LOAD DISPLACEMENT DIFFERENTIATED WITH RESPECT TO X AND INTEGRATED
C   WITH RESPECT TO Y USING A SUM OVER AN INFINITE SERIES
C   TRUNCATING THE CALCULATION AFTER THE FUNCTIONS ARE DETERMINED TO HAVE
C   CONVERGED.  THE RESULTS OF THE SIMPSON INTEGRATION AND THE INFINITE SERIES
C   INTEGRATION ARE USED TOGETHER TO CALCULATE THE NORMALIZED LOAD AT EACH JACK
C   LOCATION USING THE IMSL SUBROUTINE 'LEQT2F'.  USING THE NORMALIZED LOADS,
C   THE ACTUAL DISPLACEMENT OF THE JACKS IN THE FLOOR AND CEILING OF THE WIND
C   TUNNEL ARE CALCULATED USING MATRIX MULTIPLICATION.  THE PROGRAM THEN
C   CALCULATES THE RESIDUAL FLOW BASED ON THE DIFFERENCE BETWEEN ACTUAL
C   MEASUREMENTS OR THEORETICAL CALCULATIONS AND THE CALCULATED VALUES FROM THIS
C   PROGRAM.
C
C
C   THE ASSUMPTIONS OF THIS PROGRAM ARE AS FOLLOWS:
C             1) THE TUNNEL IS SQUARE AND THE NUMBER OF PANELS IN THE
C                Y-DIRECTON IS THE SAME AS IN THE Z-DIRECTION, AND THAT
C                DELTA_Y = DELTA_Z
C
C             2) THE NORMAL VELOCITIES ARE POSITIVE WHEN THEY FLOW INTO THE
C                TUNNEL, AND THEY ARE LOCATED AT THE CENTER OF THE PANEL.
C
C             3) THE DEFORMATIONS OF THE FLOOR AND CEILING OF THE TUNNEL ARE
C                SMALL AND CAN BE TREATED AS LINEAR.
C
C             4) THE JACKS LIE ALONG THE CENTERLINE OF THE TUNNEL (Y = 0)
C                AND THE X POSITIONS OF THE CEILING JACKS ARE THE SAME AS THE
C                FLOOR JACKS.
C
C
C   THE FOLLOWING VARIABLES ARE USED IN THE PROGRAM:
C
C      DELTA_Y = REAL, WIDTH OF PANEL IN Y DIRECTION
C      DELTA_Z = REAL, WIDTH OF PANEL IN Z DIRECTION
C      DISPL_L() = REAL, ACTUAL DISPLACEMENT OF THE LOWER JACKS AS CALCULATED
C                        BY MATRIX MULTIPLICATION
C      DISPL_U() = REAL, ACTUAL DISPLACEMENT OF THE UPPER JACKS AS CALCULATED
C                        BY MATRIX MULTIPLICATION
C      DUMMY = CHARACTER, USED TO READ COMMENT LINES IN THE INPUT FILE
C      IA = INTEGER, INITIAL DIMENSION SIZE OF THE INTEGRATION MATRIX W_INTEG()
C      IB = INTEGER, SPECIFIES THE ACCURACY OF THE ELEMENTS IN THE MATRICES
C                    SENT TO THE IMSL SUBROUTINE FOR AN ACCURACY CHECK
C                    IB = 0 INDICATES THAT AN ACCURACY CHECK IS NOT WANTED
```

```
C       IER = INTEGER, OUTPUT FROM IMSL SUBROUTINE WHICH MAY INDICATE AN ERROR
C                      IER = 129 INDICATES AN ALGORITHMICALLY SINGULAR MATRIX
C                      IER = 34 INDICATES THE RESULTS FAILED AN ACCURACY TEST
C
C       IZ = INTEGER, COMBINED NUMBER OF PANELS IN THE LOWER OR UPPER HALF OF
C                    THE TUNNEL IN THE COMBINED Y & Z DIRECTIONS, USED AS AN
C                    INDEX NUMBER
C       N = INTEGER, NUMBER OF COLUMNS IN THE MATRIX ESTABLISHED IN THE SIMPSON
C                   SUBROUTINE
C       NX = INTEGER, NUMBER OF PANELS IN THE X-DIRECTION
C       NXSI = INTEGER, NUMBER OF JACKS IN THE TUNNEL
C       NY = INTEGER, NUMBER OF PANELS IN THE Y-DIRECTION
C       NZ = INTEGER, NUMBER OF PANELS IN THE Z-DIRECTION
C       RESID_L(,) = REAL,2-D, RESIDUAL FLOW AFTER SHAPING FLOOR
C       RESID_U(,) = REAL,2-D, RESIDUAL FLOW AFTER SHAPING CEILING
C       RLENGTH = REAL, LENGTH OF WIND TUNNEL
C       SIMS_L() = REAL, INITIALLY THE INTEGRATED DISPLACEMENT (1ST INTEGRATED
C                        ONLY SPANWISE, THEN ALSO STTREAMWISE) AT EACH JACK FROM
C                        THE SIMPSON SUBROUTINE FOR THE FLOOR, AFTER IT RETURNS
C                        FROM THE IMSL SUBROUTINE IT IS THE NORMALIZED LOAD AT
C                        EACH JACK
C       SIMS_U() = REAL, INITIALLY THE INTEGRATED DISPLACEMENT (1ST INTEGRATED
C                        ONLY SPANWISE, THEN ALSO STTREAMWISE) AT EACH JACK FROM
C                        THE SIMPSON SUBROUTINE FOR THE CEILING, AFTER RETURNING
C                        FROM THE IMSL SUBROUTINE IT IS THE NORMALIZED LOAD AT
C                        EACH JACK
C       U_INF=REAL, THE FREESTREAM VELOCITY IN THE WIND TUNNEL
C       VNJ(,) = REAL, 2-D, THE NORMAL VELOCITY AT THE PANEL POSITIONS AT THE
C                           JACKS
C       VNO(,) = REAL, 2-D, THE NORMAL VELOCITY AT THE PANEL POSITIONS ON FLOOR
C                           AND CEILING
C       W_DIFF(,,) = REAL, 3-D, POINT LOAD DISPLACEMENT DIFFERENTIATED WITH
C                               RESPECT TO X, CALCULATED BY 'INFINITE SERIES'
C       W_DISP_L(,) = REAL, 2-D, POINT LOAD DISPLACEMENT CALCULATED BY 'INFINITE
C                                SERIES'
C       W_DISP_U(,) = REAL, 2-D, POINT LOAD DISPLACEMENT CALCULATED BY 'INFINITE
C                                SERIES'
C       WSTARL(,) = REAL, 2-D, SPAN INTEGRATED MATRIX, LOWER
C       WSTARU(,) = REAL, 2-D, SPAN INTEGRATED MATRIX, UPPER
C       WORKSPACE() = REAL, DIMENSIONED WORK SPACE FOR THE IMSL SUBROUTINE
C       WX_L(,) = REAL,2-D, SLOPE OF FLOOR OF WIND TUNNEL
C       WX_U(,) = REAL,2-D, SLOPE OF CEILING OF WIND TUNNEL
C       X_LOC() = REAL, LOCATION OF CENTER OF PANEL FROM THE 1/2 INCH POINT
C       XS_LOC() = REAL, X-LOCATION OF NORMAL VELOCITIES USED TO CALCULATE WSTAR
C       XSI_LOC() = REAL, LOCATION OF JACK IN X-DIRECTION RELATIVE TO THE
C                        1/2 INCH POINT
C       Y_LOC() = REAL, LOCATION OF CENTER OF PANEL IN Y-DIRECTION RELATIVE TO
C                      CENTER OF TUNNEL



        IMPLICIT REAL (A-H,O-Z)
        IMPLICIT INTEGER (I-N)

        CHARACTER DUMMY*1
        COMMON/SIMP_INT/VNJ(40,60),SIMS_U(40),SIMS_L(40)
        COMMON/LOCAT/X_LOC(100),Y_LOC(30),XSI_LOC(40),XS_LOC(40)
        COMMON/W_MATS/W_DISPL(40,40),W_DIFF(100,30,40),WSTARL(40,40),
        1      WSTARU(40,40)
        DIMENSION WORKSPACE(1360),RESID_L(100,30),RESID_U(100,30),
```

```
     1       DISPL_L(40),DISPL_U(40),WX_L(100,30),WX_U(100,30),VNO(100,40),
     2       PHINDAT(40,100)

      CALL GETCPU(NTIME1)

C   Reading information about Jack locations [VNJ.DAT]
      OPEN(2,NAME='VNJ',STATUS='OLD')
      READ(2,'(T18,I3)')NXSI
      READ(2,'(A1)')DUMMY
      READ(2,*)(XSI_LOC(I),I=1,NXSI)
      READ(2,'(A1)')DUMMY
      READ(2,*)NY,NZ
      IF(MOD(NY,2).EQ.0)THEN
          IZ=NY+NZ
          NY_2=NY/2
      ELSE
          IZ=NY+NZ+1
          NY_2=NY/2+1
      END IF
      DO I=1,5
          READ(2,'(A1)')DUMMY
      END DO
      DO I=1,NXSI
          READ(2,*)(VNJ(I,J),J=1,IZ)
      END DO
      CLOSE(2)
      PRINT*,'VNJ.DAT IS FINISHED READING'

C SET XS=XSI TO USE NORMAL VELOCITIES AT JACK LOCATIONS
      DO I=1,NXSI
        XS_LOC(I)=XSI_LOC(I)
      END DO

C Reading information about normal velocities [VNO.DAT]
      OPEN(2,NAME='VNO',STATUS='OLD')
      READ(2,'(A1)')DUMMY
      READ(2,*)RLENGTH,WIDTH
      READ(2,'(22X,F6.1)')U_INF
      READ(2,'(A1)')DUMMY
      READ(2,'(24X,I5)') NX
      READ(2,'(A1)')DUMMY
      READ(2,*)(X_LOC(J),J=1,NX)
      READ(2,'(34X,I3)')NY
      READ(2,'(18X,F6.3)')DELTA_Y
      READ(2,'(A1)')DUMMY
      READ(2,*)(Y_LOC(I),I=1,NY/2)
      READ(2,'(34X,I3)')NZ
      READ(2,'(18X,F6.3)')DELTA_Z
      IF(MOD(NY,2).EQ.0)THEN
          IZ=NY+NZ
          NY_2=NY/2
      ELSE
          IZ=NY+NZ+1
          NY_2=NY/2+1
      END IF

      DO I=1,6
        READ(2,'(A1)')DUMMY
      END DO
      DO I=1,NX
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
         READ(2,*)(VNO(I,J),J=1,IZ)
      END DO
      CLOSE(2)
      PRINT*,'VNO.DAT IS FINISHED READING'

C  Calling subroutine to calculate the integrated normal velocities
C  at each X - position specified, for the upper and lower half of
C  the wind tunnel.

      CALL SIMPSON(NXSI,NY,NZ,DELTA_Y,DELTA_Z)

C  Calling the subroutine to calculate the point load displacement matrices
C  using a summation over an infinite series, truncating when the
C  values have converged.

      CALL MATRIX(NXSI,NX,NY,RLENGTH,WIDTH)

C  The matrices have been calculated and the normal velocities have been
C  integrated, the normalized load at each jack is calculated below using
C  the IMSL subroutine [LEQT2F] to solve the matrix equation for X

C       A X = B

C  where A = the matrix of the point load displacement integrated wrt Y
C        X = the unknown normalized load vector
C        B = the normal velocities integrated over the perimeter using
C            Simpson's rule and spanwise using the trapazoiod rule


      N=1
      IER=3
      IA=40
      IB=5

C  Calling IMSL subroutine to solve for the normalized load vector for lower
C  half of tunnel

      CALL LEQT2F(WSTARL,N,NXSI,IA,SIMS_L,IB,WORKSPACE,IER)

      PRINT*,'IER =',IER                 ! print error flag to see if
      PRINT*,'IB =',IB                   ! calculations are o.k.
      IER=3        ! reset error flag for next calculation
      IB=5

C  Calling IMSL subroutine to solve for the normalized load vector for the
C  upper half of tunnel

      CALL LEQT2F(WSTARU,N,NXSI,IA,SIMS_U,IB,WORKSPACE,IER)

      PRINT*,'IER =',IER      ! print error flag to see if calculations are ok
      PRINT*,'IB =',IB
C  The variables SIMS_U(),SIMS_L() are now the normalized load vectors for the
C  jacks

      OPEN(3,NAME='LOADS',STATUS='NEW')
      WRITE(3,*)'    JACK           FLOOR            CEILING'
      WRITE(3,*)'  LOCATION         LOADS             LOADS'
      DO I=1,NXSI
          WRITE(3,*)XSI_LOC(I),SIMS_L(I),SIMS_U(I)
      END DO
```

```
        CLOSE(3)

        OPEN(3,NAME='DISPL',STATUS='NEW')
        WRITE(3,*)'    JACK          FLOOR         CEILING'
        WRITE(3,*)'  LOCATION     DISPLACEMENT    DISPLACEMENT'

C   Calculate the displacement VECTORS based on the loads
        DO I=1,NXSI
          DO J=1,NXSI
            DISPL_L(I)=DISPL_L(I)+SIMS_L(J)*W_DISPL(I,J)
            DISPL_U(I)=DISPL_U(I)+SIMS_U(J)*W_DISPL(I,J)
          END DO
          WRITE(3,*)XSI_LOC(I),DISPL_L(I),DISPL_U(I)
        END DO
        CLOSE(3)

C  Calculate the differentiated matrix (wall slope) by matrix multiplication
        OPEN(3,NAME='DWX_L',STATUS='NEW')
        OPEN(4,NAME='DWX_U',STATUS='NEW')
        DO I=1,NX
          DO J=1,NY_2
            DO K=1,NXSI
              WX_L(I,J)=WX_L(I,J)+W_DIFF(I,J,K)*SIMS_L(K)
              WX_U(I,J)=WX_U(I,J)+W_DIFF(I,J,K)*SIMS_U(K)
            END DO
          END DO
          WRITE(3,*)(WX_L(I,J),J=1,NY_2)
          WRITE(4,*)(WX_U(I,J),J=1,NY_2)
        END DO
        CLOSE(3)
        CLOSE(4)

C  Calculating the residual normal velocities for each panel on the floor
C  and ceiling of the wind tunnel
        DO I=1,NX
          DO J=1,NY_2
            RESID_L(I,J)=U_INF*WX_L(I,J)+VNO(I,J)
            JX=IZ+1-J
            RESID_U(I,J)=U_INF*WX_U(I,J)+VNO(I,JX)
          END DO
        END DO

        OPEN(2,NAME='RESIDUAL_U',STATUS='NEW')
        OPEN(3,NAME='RESIDUAL_L',STATUS='NEW')

C  Write the residuals to file [residual_u.dat] and [residual_l.dat]
        DO I=1,NX
          WRITE(2,*)(RESID_U(I,J),J=1,NY_2)
          WRITE(3,*)(RESID_L(I,K),K=1,NY_2)
        END DO
        CLOSE(2)
        CLOSE(3)

C  Create data for FHIXZM.FOR program
        DO I=1,NY_2
          DO J=1,NX
            FHINDAT(I,J)=RESID_L(J,I)
          END DO
        END DO
```

```
      DO I=NY_2+1,NY_2+NZ
          DO J=1,NX
              PHINDAT(I,J)=VNO(J,I)
          END DO
      END DO

      DO I=NY2+NZ+1,NY+NZ
          DO J=1,NX
              K=IZ+1-I
              PHINDAT(I,J)=RESID_U(J,K)
          END DO
      END DO

C  Write data to file in format to be read by PHIXZM.FOR program
      OPEN(3,NAME='PHINWALL',STATUS='NEW')
      WRITE(3,*)((PHINDAT(I,J),I=1,IZ),J=1,NX)
      CLOSE(3)


      CALL GETCPU(NTIME2)
      TIME=(NTIME2-NTIME1)/100.
      NMIN=INT(TIME/60.)
      NSEC=INT(AMOD(TIME,60.))
      PRINT'(A,I3,A,I3,A)','CPU RUN TIME =',NMIN,' MINUTES',NSEC,' SEC
      1ONDS'
2000  STOP 'EXITING PROGRAM'
      END




      SUBROUTINE MATRIX(NXSI,NX,NY,RLENGTH,WIDTH)

C  THIS SUBROUTINE CALCULATES THE MATRICES FOR THE DISPLACEMENT, THE
C  DISPLACEMENT DIFFERENTIATED WITH RESPECT TO X, AND THE SPANWISE
C  INTEGRATED DISPLACEMENT.  THE CALCULATED VALUES ARE STORED
C  IN A COMMON BLOCK LABELED 'W_MATS', AND THE VARIABLES USED IN THE
C  CALCULATIONS ARE DEFINED BELOW:

C  THE FOLLOWING VARIABLES ARE USED IN THE PROGRAM:
C       X_LOC() =  THE X - DISTANCE ALONG THE LENGTH OF THE WIND TUNNEL
C       Y_LOC() =  THE Y - DISTANCE ALONG THE WIDTH OF THE TUNNEL (FROM CENTER)
C       NX = THE NUMBER OF X LOCATIONS TO BE USED FOR CALCULATION
C       NXSI = THE NUMBER OF XSI (JACK) LOCATIONS TO BE USED FOR CALCULATIONS
C       NY = THE NUMBER OF Y LOCATIONS TO BE USED FOR THE CALCULATION
C       XSI_LOC() = LOCATION OF THE JACKS ALONG THE LENGTH OF THE WIND TUNNEL
C       RLENGTH = TOTAL LENGTH OF THE WIND TUNNEL
C       WIDTH = TOTAL WIDTH OF THE WIND TUNNEL
C       W_DISP_L(,) = ELEMENT IN THE 2-D DISPLACEMENT MATRIX
C       W_DISP_U(,) = ELEMENT IN THE 2-D DISPLACEMENT MATRIX
C       W_DIFF(,,) = ELEMENT IN THE 3-D DIFFERENTIATED MATRIX
C       W_INTEG(,) = ELEMENT IN THE 2-D INTEGRATED LOWER MATRIX
C       ALPHA_M = PARAMETER USED IN CALCULATING THE ABOVE
C               = M*PI*WIDTH/(RLENGTH*2.)
C          WHERE:
C                  M = INDEXING VALUE FOR THE INFINITE SUM
C                  PI = 3.41592654
C       B = A RELATIVE Y - DISTANCE BETWEEN 0 and 1
```

```
        IMPLICIT REAL (A-H,O-Z)
        IMPLICIT INTEGER (I-N)

        COMMON/LOCAT/ X_LOC(100),Y_LOC(30),XSI_LOC(40),XS_LOC(40)
        COMMON/W_MATS/W_DISPL(40,40),W_DIFF(100,30,40),WSTARL(40,40),
        1       WSTARU(40,40)

        PARAMETER (PI=3.141592654)
        CONST=PI*WIDTH/(2.*RLENGTH)

        OPEN(3,NAME='DISP_MAT',STATUS='NEW')

C  Calculation of the displacement matrix follows below
        FACTOR=2.*PI**3
        DO I=1,NXSI
            DO J=1,NXSI
                DO M=1,150
                   ALPHA_M=M*CONST
                   W1=((1.+ALPHA_M*TANH(ALPHA_M))*TANH(ALPHA_M)-ALPHA_M)*
        1             SIN(M*PI*XSI_LOC(J)/RLENGTH)*SIN(M*PI*XSI_LOC(I)/RLENGTH)
        2            /M**3
                   W_DISPL(I,J)=W_DISPL(I,J)+W1

C AN AVERAGING OF THE LAST TERMS OF THE SERIES IS EMPLOYED FOR SMOOTHING
                   T10=T9
                   T9=T8
                   T8=T7
                   T7=T6
                   T6=T5
                   T5=T4
                   T4=T3
                   T3=T2
                   T2=T1
                   T1=W_DISPL(I,J)
                END DO
100             W_DISPL(I,J)=(T1+T2+T3+T4+T5+T6+T7+T8+T9+T10)/(10.*FACTOR)
            END DO
            WRITE(3,*)(W_DISPL(I,J),J=1,NXSI)
        END DO
        CLOSE(3)
        PRINT*,'DISPLACEMENT MATRIX IS CALCULATED'


C  Calculation of the differentiated matrix follows below!
        FACTOR=RLENGTH*2.*PI**2
        DO I=1,NX
            DO J=1,NY/2
                DO K=1,NXSI
                    B=(WIDTH-2.*Y_LOC(J))/WIDTH
                    DO M=1,150
                       ALPHA_M=M*CONST
                       IF(B*ALPHA_M.GT.10.)THEN
                           W2=((1.+ALPHA_M*2.*Y_LOC(J)/WIDTH)*EXP(-ALPHA_M
        1                       *2.*Y_LOC(J)/WIDTH))*SIN(M*PI*XSI_LOC(K)/RLENGTH)
        2                       *COS(M*PI*X_LOC(I)/RLENGTH)/M**2
                       ELSE
                           W2=((1.+ALPHA_M*TANH(ALPHA_M))*SINH(ALPHA_M*B)-
        1                       ALPHA_M*B*COSH(ALPHA_M*B))*SIN(M*PI*XSI_LOC(K)/
        2                       RLENGTH)*COS(M*PI*X_LOC(I)/RLENGTH)/(M**2*
        3                       COSH(ALPHA_M))
```

```
                 END IF
                 W_DIFF(I,J,K)=W_DIFF(I,J,K)+W2
               END DO
200         W_DIFF(I,J,K)=W_DIFF(I,J,K)/FACTOR
            END DO
         END DO
         PRINT*,I
      END DO
      PRINT*,'DIFFERENTIATED MATRIX IS CALCULATED'

C   Calculation of spanwise integrated matrix begins below
      OPEN(3,NAME='WSTAR_MAT',STATUS='NEW')
      FACTOR=RLENGTH/(PI**4)
      DO I=1,NXSI
         DO J=1,NXSI
            WSTARL(I,J)=0.
            DO M=1,120
               ALPHA_M=M*CONST
               W3=SIN(M*PI*XSI_LOC(J)/RLENGTH)*SIN(M*PI*XS_LOC(I)/
      1           RLENGTH)*(2.-(2.+ALPHA_M*TANH(ALPHA_M))/
      2           COSH(ALPHA_M))/M**4
               WSTARL(I,J)=WSTARL(I,J)+W3
               T10=T9
               T9=T8
               T8=T7
               T7=T6
               T6=T5
               T5=T4
               T4=T3
               T3=T2
               T2=T1
               T1=WSTARL(I,J)
            END DO
         WSTARL(I,J)=(T1+T2+T3+T4+T5+T6+T7+T8+T9+T10)/10.*FACTOR
         WSTARU(I,J)=WSTARL(I,J)
         END DO
         WRITE(3,*)(WSTARL(I,J),J=1,NXSI)
      END DO

C***  Debugging Diagnostics
      DO I=1,NXSI
        WRITE(3,*)I,XSI_LOC(I),XS_LOC(I)
      END DO
      CLOSE(3)

      PRINT*,'WSTAR MATRIX IS CALCULATED'

      RETURN
      END



      SUBROUTINE SIMPSON(NXSI,NY,NZ,DELTA_Y,DELTA_Z)


      IMPLICIT REAL (A-H,O-Z)
      IMPLICIT INTEGER (I-N)
      INTEGER START

      COMMON/SIMP_INT/VNJ(40,60),SIMS_U(40),SIMS_L(40)
```

```
        COMMON/LOCAT/X_LOC(100),Y_LOC(30),XSI_LOC(40),XS_LOC(40)

        DIMENSION VINTXU(40),VINTXL(40)

C  CHECK TO SEE DELTA_Z AND DELTA_Y ARE THE SAME, IF THEY ARE, USE SIMPSON'S
C  INTEGRATION AROUND IZ, IF NOT, INTEGRATE EACH SEGMENT OF NY AND NZ SEPERATLY
        IF(ABS(DELTA_Y-DELTA_Z).GT..001)GOTO 100



*************************************************************************
C THIS PORTION IS FOR EQUAL PANEL WIDTHS IN THE Y AND Z DIRECTIONS
C  CHECK TO SEE IF NY IS ODD OR EVEN FOR THE APPROPRIATE END CORRECTIONS
        IF(MOD(NY,2).NE.0)THEN
            NYFLAG=1
            IZ=NY+NZ+1
            IZ_2=IZ/2
        ELSE
            NYFLAG=0
            IZ=NY+NZ
            IZ_2=IZ/2
        END IF

C  SIMPSON'S INTEGRATION ROUTINE FOR  DELTA_Z = DELTA_Y
        DO KX=1,NXSI     ! EACH X LOCATION
        START=2
C  CHECK FOR ODD OR EVEN INTERVAL, IF ODD, APPLY SIMPSON'S 3/8 RULE
C  FOR THE FIRST FOUR POINTS
            IF(MOD(IZ_2,2).EQ.0)THEN
                SIMS_L(KX)=3.*DELTA_Z/8.*(VNJ(KX,1)+3.*VNJ(KX,2)+3.*
        1       VNJ(KX,3)+VNJ(KX,4))
                START=5
            END IF

C  START OF SIMPSON'S INTEGRATION FOR EVEN INTERVALS
            IF(MOD(START,2).NE.0)THEN
                IEOFLAG=1
            ELSE
                IEOFLAG=0
            END IF
            DO KZ=START,IZ_2-1
                IF(MOD(KZ,2).EQ.0)THEN
                    IF(IEOFLAG.EQ.1)THEN
                        SUM=SUM+2.*VNJ(KX,KZ)
                    ELSE
                        SUM=SUM+4.*VNJ(KX,KZ)
                    END IF
                ELSE
                    IF(IEOFLAG.EQ.1)THEN
                        SUM=SUM+4.*VNJ(KX,KZ)
                    ELSE
                        SUM=SUM+2.*VNJ(KX,KZ)
                    END IF
                END IF
            END DO

            SIMS_L(KX)=SIMS_L(KX)+(SUM+VNJ(KX,START-1)+VNJ(KX,IZ_2))
        1   *DELTA_Z/3.
            SUM=0.
            END_VN=(VNJ(KX,IZ_2)+VNJ(KX,IZ_2+1))/2.
```

```
C  CHECK FOR WHICH END CORRECTION TO USE
          IF(NYFLAG.EQ.1)THEN
             ENDCORR=DELTA_Z/2.*(MIN(VNJ(KX,IZ_2),END_VN)+ABS(
     1       VNJ(KX,IZ_2)-END_VN)/2.)
          ELSE
             ENDCORR=DELTA_Z/2.*(VNJ(KX,1)+MIN(VNJ(KX,IZ_2),
     1       END_VN)+ABS(VNJ(KX,IZ_2)-END_VN)/2.)
          END IF
C  INTEGRATED VALUE FOR THE LOWER HALF OF TUNNEL
          SIMS_L(KX)=-2.*(SIMS_L(KX)+ENDCORR)

C  START SIMPSON'S ROUTINE FOR THE UPPER HALF OF TUNNEL
          START=IZ_2+2

C  CHECK FOR ODD OR EVEN INTERVAL
          IF(MOD(IZ_2,2).EQ.0)THEN
             SIMS_U(KX)=3.*DELTA_Z/8.*(VNJ(KX,IZ_2+1)+3.*
     1       VNJ(KX,IZ_2+2)+3.*VNJ(KX,IZ_2+3)+VNJ(KX,IZ_2+4))
             START=IZ_2+5
          END IF

C  START THE INTEGRATION FOR EVEN INTERVAL
          IF(MOD(START,2).NE.0)THEN
             IEOFLAG=1
          ELSE
             IEOFLAG=0
          END IF
          DO KZ=START,IZ-1
             IF(MOD(KZ,2).EQ.0)THEN
                IF(IEOFLAG.EQ.1)THEN
                   SUM=SUM+2.*VNJ(KX,KZ)
                ELSE
                   SUM=SUM+4.*VNJ(KX,KZ)
                END IF
             ELSE
                IF(IEOFLAG.EQ.1)THEN
                   SUM=SUM+4.*VNJ(KX,KZ)
                ELSE
                   SUM=SUM+2.*VNJ(KX,KZ)
                END IF
             END IF
          END DO
          SIMS_U(KX)=SIMS_U(KX)+DELTA_Z*(VNJ(KX,START-1)+SUM+
     1    VNJ(KX,IZ))/3.
          SUM=0.
          END_VN=(VNJ(KX,IZ_2)+VNJ(KX,IZ_2+1))/2.
C  CHECK TO SEE WHICH END CORRECTION APPLIES
          IF(NYFLAG.EQ.1)THEN
             ENDCORR=DELTA_Z/2.*(MIN(VNJ(KX,IZ_2+1),END_VN)+
     1       ABS(VNJ(KX,IZ_2+1)-END_VN)/2.)
          ELSE
             ENDCORR=DELTA_Z/2.*(VNJ(KX,IZ)+MIN(VNJ(KX,IZ_2+1),
     1       END_VN)+ABS(VNJ(KX,IZ_2+1)-END_VN)/2.)
          END IF
C  INTEGRATED VALUE FOR THE UPPER HALF OF THE TUNNEL
          SIMS_U(KX)=-2.*(SIMS_U(KX)+ENDCORR)
       END DO

       GOTO 1000      ! CALCULATIONS ARE COMPLETE, WRITE TO FILE
```

```
*************************************************************************
C THIS SEGMENT FOR PANELS OF UNEQUAL WIDTHS IN THE Y AND Z DIRECTION
C  SIMPSON'S ROUTINE FOR DELTA_Y .NE. DELTA_Z

100     IF(MOD(NY,2).EQ.0)THEN
            NYFLAG=0
            NY_2=NY/2
        ELSE
            NYFLAG=1
            NY_2=NY/2+1
        END IF

        NZ_2=NZ/2

C  SIMPSON'S ROUTINE FOR  DELTA_Z .NE. DELTA_Y
        IEOFLAG=0
        DO KX=1,NXSI
        START=2
            IF(MOD(NY_2,2).EQ.0)THEN
                SIMS_LY=3.*DELTA_Y/8.*(VNJ(KX,1)+3.*VNJ(KX,2)+3.*
        1       VNJ(KX,3)+VNJ(KX,4))
                START=5
            END IF

            IF(MOD(START,2).NE.0)THEN
                IEOFLAG=1
            ELSE
                IEOFLAG=0
            END IF
            DO KZ=START,NY_2-1
                IF(MOD(KZ,2).EQ.0)THEN
                    IF(IEOFLAG.EQ.1)THEN
                        SUM=SUM+2.*VNJ(KX,KZ)
                    ELSE
                        SUM=SUM+4.*VNJ(KX,KZ)
                    END IF
                ELSE
                    IF(IEOFLAG.EQ.1)THEN
                        SUM=SUM+4.*VNJ(KX,KZ)
                    ELSE
                        SUM=SUM+2.*VNJ(KX,KZ)
                    END IF
                END IF
            END DO

            SIMS_LY=SIMS_LY+(SUM+VNJ(KX,START-1)+VNJ(KX,NY_2))
        1   *DELTA_Y/3.
            SUM=0.
            END_VN=(VNJ(KX,NY_2)+VNJ(KX,NY_2+1))/2.
            IF(NYFLAG.EQ.1)THEN
                ENDCORR=DELTA_Y/2.*(MIN(VNJ(KX,NY_2),END_VN)+
        1       ABS(VNJ(KX,NY_2)-END_VN)/2.)
            ELSE
                ENDCORR=DELTA_Y/2.*(VNJ(KX,1)+MIN(VNJ(KX,NY_2),
        1       END_VN)+ABS(VNJ(KX,NY_2)-END_VN)/2.)
            END IF
            SIMS_LY=SIMS_LY+ENDCORR

            START=NY_2+2
            IF(MOD(NZ_2,2).EQ.0)THEN
```

```
            SIMS_LZ=3.*DELTA_Z/8.*(VNJ(KX,NY_2+1)+3.*
     1      VNJ(KX,NY_2+2)+3.*VNJ(KX,NY_2+3)+VNJ(KX,NY_2+4))
            IEOFLAG=1
            START=NY_2+5
         END IF

         IF(MOD(START,2).NE.0)THEN
            IEOFLAG=1
         ELSE
            IEOFLAG=0
         END IF
         DO KZ=START,NY_2+NZ_2-1
            IF(MOD(KZ,2).EQ.0)THEN
               IF(IEOFLAG.EQ.1)THEN
                  SUM=SUM+2.*VNJ(KX,KZ)
               ELSE
                  SUM=SUM+4.*VNJ(KX,KZ)
               END IF
            ELSE
               IF(IEOFLAG.EQ.1)THEN
                  SUM=SUM+4.*VNJ(KX,KZ)
               ELSE
                  SUM=SUM+2.*VNJ(KX,KZ)
               END IF
            END IF
         END DO

         SIMS_LZ=SIMS_LZ+(SUM+VNJ(KX,START-1)+VNJ(KX,NY_2+NZ_2))
     1   *DELTA_Z/3.
         SUM=0.
         END_VN1=(VNJ(KX,NY_2)+VNJ(KX,NY_2+1))/2.
         END_VN2=(VNJ(KX,NY_2+NZ_2)+VNJ(KX,NY_2+NZ_2+1))/2.
         ENDCORR1=DELTA_Z/2.*(MIN(VNJ(KX,NY_2+1),END_VN1)+
     1   ABS(VNJ(KX,NY_2+1)-END_VN1)/2.)
         ENDCORR2=DELTA_Z/2.*(MIN(VNJ(KX,NY_2+NZ_2),END_VN2)+
     1   ABS(VNJ(KX,NY_2+NZ_2)-END_VN2)/2.)
         SIMS_L(KX)=-2.*(SIMS_LY+SIMS_LZ+ENDCORR1+ENDCORR2)
         SIMS_LY=0.
         SIMS_LZ=0.

         START=NY_2+NZ_2+2
         IF(MOD(NZ_2,2).EQ.0)THEN
            SIMS_UZ=3.*DELTA_Z/8.*(VNJ(KX,NY_2+NZ_2+1)+3.*
     1      VNJ(KX,NY_2+NZ_2+2)+3.*VNJ(KX,NY_2+NZ_2+3)+
     2      VNJ(KX,NY_2+NZ_2+4))
            START=NY_2+NZ_2+5
         END IF


         IF(MOD(START,2).NE.0)THEN
            IEOFLAG=1
         ELSE
            IEOFLAG=0
         END IF
         DO KZ=START,NY_2+NZ-1
            IF(MOD(KZ,2).EQ.0)THEN
               IF(IEOFLAG.EQ.1)THEN
                  SUM=SUM+2.*VNJ(KX,KZ)
               ELSE
                  SUM=SUM+4.*VNJ(KX,KZ)
```

```
            END IF
        ELSE
            IF(IEOFLAG.EQ.1)THEN
                SUM=SUM+4.*VNJ(KX,KZ)
            ELSE
                SUM=SUM+2.*VNJ(KX,KZ)
            END IF
        END IF
    END DO

    SIMS_UZ=SIMS_UZ+(SUM+VNJ(KX,START-1)+VNJ(KX,NY_2+NZ))
1   *DELTA_Z/3.
    SUM=0.
    END_VN2=(VNJ(KX,NY_2+NZ)+VNJ(KX,NY_2+NZ+1))/2.
    END_VN1=(VNJ(KX,NY_2+NZ_2)+VNJ(KX,NY_2+NZ_2+1))/2.
    ENDCORR2=DELTA_Z/2.*(MIN(VNJ(KX,NY_2+NZ),END_VN2)+
1   ABS(VNJ(KX,NY_2+NZ)-END_VN2)/2.)
    ENDCORR1=DELTA_Z/2.*(MIN(VNJ(KX,NY_2+NZ_2+1),END_VN1)+
1   ABS(VNJ(KX,NY_2+NZ_2+1)-END_VN1)/2.)
    SIMS_UZ=SIMS_UZ+ENDCORR1+ENDCORR2

START=NY_2+NZ+2
    IF(MOD(NY_2,2).EQ.0)THEN
        SIMS_UY=3.*DELTA_Y/8.*(VNJ(KX,NY_2+NZ+1)+3.*
1       VNJ(KX,NY_2+NZ+2)+3.*VNJ(KX,NY_2+NZ+3)+
2       VNJ(KX,NY_2+NZ+4))
        START=NY_2+NZ+5
    END IF

    IF(MOD(START,2).NE.0)THEN
        IEOFLAG=1
    ELSE
        IEOFLAG=0
    END IF
    DO KZ=START,IZ-1
        IF(MOD(KZ,2).EQ.0)THEN
            IF(IEOFLAG.EQ.1)THEN
                SUM=SUM+2.*VNJ(KX,KZ)
            ELSE
                SUM=SUM+4.*VNJ(KX,KZ)
            END IF
        ELSE
            IF(IEOFLAG.EQ.1)THEN
                SUM=SUM+4.*VNJ(KX,KZ)
            ELSE
                SUM=SUM+2.*VNJ(KX,KZ)
            END IF
        END IF
    END DO

    SIMS_UY=SIMS_UY+(SUM+VNJ(KX,START-1)+VNJ(KX,IZ))
1   *DELTA_Y/3.
    SUM=0.
    END_VN=(VNJ(KX,NY_2+NZ)+VNJ(KX,NY_2+NZ+1))/2.
    IF(NYFLAG.EQ.1)THEN
        ENDCORR=DELTA_Y/2.*(MIN(VNJ(KX,NY_2+NZ+1),END_VN)+
1       ABS(VNJ(KX,NY_2+NZ+1)-END_VN)/2.)
    ELSE
        ENDCORR=DELTA_Y/2.*(VNJ(KX,IZ)+MIN(VNJ(KX,
1       NY_2+NZ+1),END_VN)+ABS(VNJ(KX,NY_2+NZ+1)-END_VN)/2.)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
           END IF
           SIMS_U(KX)=-2.*(SIMS_UZ+SIMS_UY+ENDCORR)
           SIMS_UZ=0.
           SIMS_UY=0.
        END DO

1000    OPEN(3,NAME='SIMSINTEG',STATUS='NEW')
        WRITE(3,*)NXSI
        DO I=1,NXSI
        WRITE(3,*)XSI_LOC(I),SIMS_L(I),SIMS_U(I)
        END DO

C*************************************************

C COMPUTE THE STREAMWISE INTEGRAL OF THESE SPAN INTEGRATED VELOCITIES
C       AS A FUNCTION OF X
C      THIS IS USED SOLUTION FOR THE JACK LOADS BY USE OF THE WSTAR
C       (SPAN INTEGRATED DISPLACEMENT) MATRIX


        VINTXL(1) = 0.5*SIMS_L(1)* XS_LOC(1)
        VINTXU(1) = 0.5*SIMS_U(1)* XS_LOC(1)
C  THIS IMPLICITLY ASSUMES THAT SIMS(X=0) = 0

        DO I=2,NXSI
                VINTXL(I)=VINTXL(I-1) + 0.5*(SIMS_L(I)+SIMS_L(I-1)) *
     *                  (XS_LOC(I)-XS_LOC(I-1))
                VINTXU(I)=VINTXU(I-1) + 0.5*(SIMS_U(I)+SIMS_U(I-1)) *
     *                  (XS_LOC(I)-XS_LOC(I-1))
                END DO
C  DUMP STREAMWISE INTEGRATED VELOCITIES BACK INTO SIMS_ ARRAYS

        DO I=1,NXSI
                SIMS_U(I)=VINTXU(I)
                SIMS_L(I)=VINTXL(I)
                END DO

        WRITE(3,3000)
 3000   FORMAT(X,'FOLLOWING ARE THE STREAMWISE INTEGRATED VELOCITIES')

        WRITE(3,*)NXSI
        DO I=1,NXSI
        WRITE(3,*)XSI_LOC(I),XS_LOC(I),SIMS_L(I),SIMS_U(I)
        END DO

        CLOSE(3)
        PRINT*,'SIMPSON''S INTEGRATION COMPLETE'
        RETURN
        END
```

```
                  PROGRAM AFMODLJ

C           COMPUTES THE WALL- FREE VELOCITIES(VN) NORMAL TO THE
C           WALLS OF A SQUARE WIND TUNNEL OF AN AEDC MODEL
C           TO BE USED IN A LANGLEY EXPERIMENTAL TEST.
C           IT ALSO COMPUTES DATA AT THE 0.3 METER TUNNEL JACKS AND
C           WRITES THE FILES 'VNO.DAT' AND 'VNJ.DAT' FOR JACK_DISPL.
C           THIS PROGRAM USES 2*NX+1 PLUS DX PANELS OVER THE LENGTH OF
C           THE WIND TUNNEL AND NY+NZ PANELS AROUND THE HALF
C           PERIMETER.  TO INCREASE THE NUMBER OF PANELS
C           NEW VALUES MAY BE NEEDED IN THE DIMENSION STATEMENT
C           AND THE LIMITS NX, DX AND NZ .

C           THIS PROGRAM REPRESENTS THE MODEL BY TWO SOURCES(BODY)
C           VEE LINE-VORTICES(WING&TAIL LIFT) AND +&- LINE SOURCES
C           (WING&TAIL THICKNESS).BODY LIFT BY LINE DOUBLETS
C           IT ASSUMES THE CP IS LOCATED AT THE ZERO POSITION OF THE
C            X COORDINATES

C           THE WING IS ASSUMED TO CARRY 80% OF THE TOTAL LIFT

            DIMENSION X(-80:120),XJ(-20:0),Y(80),Z(80),VNS(80,-80:120),
     1        VNA(80,-80:120),VN(80,-80:120),DZXL(-80:120),DZXU(-80:120),
     1        ZL(-80:120),ZU(-80:120),VNN(80,-80:120),VNJ(80,-80:120),
     1        X_LOC(-80:120)

C           XJ(IX) ARE THE 1/3 METER TUNNEL JACK LOCATIONS
C           X(IX),Y(IZ) AND Z(IZ) ARE THE CENTERS OF THE PANELS

C           THE TRACOR ALGORITHM IS APPLIED IN THIS PROGRAM
C           THE NUMBERS BELOW ARE FIXED BY THE DIMENSIONS OF THE
C           AEDC MODEL

            PARAMETER (PI=3.14159,S=0.10668, SV=.0837863,AR=3.5,
     1        D=0.03556,C=.060198,CT=0.03556,WT=.0072238,WTT=.004267,
     1        ST=.05588,SVT=.043888,A=.33, ART=3.1429)
            REAL  MACH
            TYPE 10
10          FORMAT(10X,'ENTER  CL  MACH  NX   NZ')

            READ(5,*)  CL, MACH, NX, NZ
            BETA=SQRT(1-MACH*MACH)

C           THE FOLLOWING RESULT FROM THE MODEL DIMENSIONS

            XRVC=-.03759     !LOCATION OF VORTEX &C/L INTERSECTION(METERS)
            XRV=XRVC/BETA
            XRVTC=0.0889  !SAME FOR THE TAIL
            XRVT=XRVTC/BETA
            XSL=-.10795        !LOCATION OF FOREWARD SOURCE
            XST=0.1524            ! METERS, SAME FOR THE REAR SOURCE
            TG=.57735/BETA  ! TANGENT OF THE STRETCHED SWEEPANGLE
C                            REAL SWEEP IS 30 DEGREES!

            TS=1+TG*TG

            DX=0
            NZ2=NZ/2
            NY-NZ
            NY2=NY/2
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C             NOW SET UP THE COORDINATES OF THE POINTS AT WHICH
C               VELOCITIES ARE TO BE CALCULATED.

              DO IZ=1,NZ2
                Z(IZ)=A/2
                Y(IZ)=A/NZ*(IZ-0.5)
              END DO
              DO IZ=NZ2+1,NY2+NZ
                Y(IZ)=A/2
                Z(IZ)=A/NZ*(NZ+0.5-IZ)
              END DO
              DO IZ=NZ+NY2+1,NZ+NY
                Z(IZ)=-A/2
                Y(IZ)=(NZ+NY+0.5-IZ)*A/NZ
              END DO
              DO IX=-NX,NX+DX
                X(IX)=A/NZ*IX   ! FOR PANEL CENTERS RELATIVE TO THE CP POINT
              END DO
              XJ(-20)=-23.25*.0254
              XJ(-19)=-17.5*.0254
              XJ(-18)=-12.5*.0254
              XJ(-17)=-8.5*.0254
              XJ(-16)=-5.5*.0254
              XJ(-15)=-3.5*.0254
              XJ(-14)=-2.0*.0254
              XJ(-13)=-0.5*.0254
              XJ(-12)=1.0*.0254
              XJ(-11)=2.5*.0254
              XJ(-10)=4.0*.0254
              XJ(-9)=5.5*.0254
              XJ(-8)=7.5*.0254
              XJ(-7)=9.5*.0254
              XJ(-6)=11.5*.0254
              XJ(-5)=14.5*.0254
              XJ(-4)=18.5*.0254
              XJ(-3)=23.5*.0254
              XJ(-2)=28.5*.0254
              XJ(-1)=33.5*.0254
              XJ(0)=39.0*.0254

              CIRC=0.8*CL*S*S/(SV*AR)
              ALPHA=0.8*CL/(3.2+1.755*MACH**6)   !IN RADIANS
C             THIS ASSUMES AN APPROXIMATE MACH DEPENCENCE FUR CL/ALPHA

              Q=WT
              QT=WTT
              CIRCT=.2*CL*ST*ST/(SVT*ART)  !TAIL CIRCULATION
              DDD=1+.5*D*D*BETA*BETA/((XST-XSL)*(XST-XSL))  !APPROX STRENGTH CORRECTOR

              DO IX=-NX,NX+DX

C             STRETCHED DISTANCES FROM FIELD POINT TO INTERSECTION OF C/L AND:
                XV=(XRVC-X(IX))/BETA         ! WING VORTEX
                XL=(XRVC-C/4-X(IX))/BETA     ! WING LEADING EDGE
                XT=(XRVC+3*C/4-X(IX))/BETA   ! WING TRAILING EDGE
                XLS=(XSL-X(IX))/BETA         ! BODY SOURCE
                XTS=(XST-X(IX))/BETA         ! BODY SINK(AFT)
                XVT=(XRVTC-X(IX))/BETA       ! TAIL VORTEX
                XLT=XVT-CT/4/BETA            ! TAIL LEADING EDGE
```

```
        XTT=XVT+3*CT/4/BETA        ! TAIL TRAILING EDGE

C       RM IS THE ARM LENGTH FROM THE POINT OF ROTATION OF
C       THE BALANCE/STING SYSTEM TO THE CP ORIGIN OF COORDS.
C       THE MODEL IS DISPLACED UPWARDS BY A DISTANCE ALPHA*RM

        RM=0.501269 !METERS

        DO IZ=1,N/2
          F=(XV+Y(IZ)*TG)
          F1=(XV-Y(IZ)*TG)
          FT=(XVT+Y(IZ)*TG)
          FT1=(XVT-Y(IZ)*TG)
          YY=Y(IZ)**2
          ZZ=(Z(IZ)+ALPHA*RM)**2
          Q1=4*(F*F+TS*ZZ)
          Q2=4*(F1*F1+TS*ZZ)
          Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
          Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
          Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
          Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
          Q1T=4*(FT*FT+TS*ZZ)
          Q2T=4*(F1T*F1T+TS*ZZ)
          Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
          Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
          Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
          Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
          R1=SQRT(XV**2+ZZ+YY)
          R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
          R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
          R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
          R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
          R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
          R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
          R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
          R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
          R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
          R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
          R7=ZZ+YY+XLS*XLS
          R8=ZZ+YY+XTS*XTS

          R1T=SQRT(XVT**2+ZZ+YY)
          R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
          R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
          R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
          R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
          R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
          R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
          R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
          R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
          R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
          R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)


        VNS(IZ,IX)=-D*D*DDD*(Z(IZ)+ALPHA*RM)/16/BETA*(1/(R7*SQRT(R7))
     1    -0.61/(R8*SQRT(R8)))
C       THESE ABOVE ARE THE BODY SOURCE TERMS
        VNS(IZ,IX)=-Q*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*S+XL*TG-Y(IZ))/(Q3*R4)-
     1    (TS*D/2+XL*TG-Y(IZ))/(Q3*R3)+(TS*S+XL*TG+Y(IZ))/(Q4*R41)-
     1    (TS*D/2+XL*TG+Y(IZ))/(Q4*R31)-(TS*S+XT*TG-Y(IZ))/(Q5*R6)+
```

```
     1          (TS*D/2+XT*TG-Y(IZ))/(Q5*R5)-(TS*S+XT*TG+Y(IZ))/(Q6*R61)+
     1          (TS*D/2+XT*TG+Y(IZ))/(Q6*R51))+VNS(IZ,IX)
C             THESE WERE THE WING LINE SOURCE TERMS

              VNS(IZ,IX)=-QT*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*ST+XLT*TG-Y(IZ))/
     1          (Q3T*R4T)-(TS*D/2+XLT*TG-Y(IZ))/(Q3T*R3T)+(TS*ST+XLT*TG+Y(IZ))/
     1          (Q4T*R41T)-(TS*D/2+XLT*TG+Y(IZ))/(Q4T*R31T)-(TS*ST*XTT*TG-
     1          Y(IZ))/(Q5T*R6T)+(TS*D/2+XTT*TG-Y(IZ))/(Q5T*R5T)-
     1          (TS*ST+XTT*TG+Y(IZ))/(Q6T*R61T)+(TS*D/2+XTT*TG+Y(IZ))/
     1          (Q6T*R51T))+VNS(IZ,IX)
C             THESE WERE THE TAIL LINE SOURCE TERMS

              VNA(IZ,IX)=D*D*ALPHA/8*((YY-ZZ)/(YY+ZZ)**2*(1-XLS/SQRT(R7))
     1         +ZZ*XLS/(YY+ZZ)/(R7*SQRT(R7))-0.61*((YY-ZZ)/(YY+ZZ)**2*
     1         (1-XTS/SQRT(R8))+ZZ*XTS/(YY+ZZ)/(R8*SQRT(R8))))
C             THESE ARE THE DOUBLET(BODY LIFT) TERMS


              TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
              TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

              VNA(IZ,IX)=VNA(IZ,IX)+CIRC/4/PI*(F*TT+F1*TO+(Y(IZ)-SV)/
     1          (ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)-(Y(IZ)+SV)/(ZZ+
     1           (Y(IZ)+SV)**2)*(1-(XV+SV*TG)/R21))
C             THE ABOVE ARE THE WING LINE VORTEX TERMS

              TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
              TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

              VNA(IZ,IX)=VNA(IZ,IX)+CIRCT/4/PI*(FT*TTT+F1T*TOT+(Y(IZ)-SVT)/
     1          (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)-(Y(IZ)+SVT)/
     1          (ZZ+(Y(IZ)+SVT)**2)*(1-(XVT+SVT*TG)/R21T))
C             THE ABOVE ARE THE TAIL VORTEX TERMS

              VN(IZ,IX)=VNA(IZ,IX)+VNS(IZ,IX)

              END DO

              DO IZ=NY2+NZ+1,NY+NZ    !UPPER WALL
                F=(XV+Y(IZ)*TG)
                F1=(XV-Y(IZ)*TG)
                FT=(XVT+Y(IZ)*TG)
                FT1=(XVT-Y(IZ)*TG)
                YY=Y(IZ)**2
                ZZ=(Z(IZ)+ALPHA*RM)**2
                Q1=4*(F*F+TS*ZZ)
                Q2=4*(F1*F1+TS*ZZ)
                Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
                Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
                Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
                Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
                Q1T=4*(FT*FT+TS*ZZ)
                Q2T=4*(F1T*F1T+TS*ZZ)
                Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
                Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
                Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
                Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
                R1=SQRT(XV**2+ZZ+YY)
                R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
                R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
```

```
         R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
         R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
         R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
         R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
         R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
         R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
         R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
         R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
         R7=ZZ+YY+XLS*XLS
         R8=ZZ+YY+XTS*XTS

         R1T=SQRT(XVT**2+ZZ+YY)
         R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
         R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
         R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
         R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
         R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
         R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
         R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
         R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
         R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
         R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)


      VNS(IZ,IX)=+D*D*DDD*(Z(IZ)+ALPHA*RM)/16/BETA*(1/(R7*SQRT(R7))
     1  -0.61/(R8*SQRT(R8)))
C     THESE ABOVE ARE THE BODY SOURCE TERMS
      VNS(IZ,IX)=Q*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*S+XL*TG-Y(IZ))/(Q3*R4)-
     1  (TS*D/2+XL*TG-Y(IZ))/(Q3*R3)+(TS*S+XL*TG+Y(IZ))/(Q4*R41)-
     1  (TS*D/2+XL*TG+Y(IZ))/(Q4*R31)-(TS*S+XT*TG-Y(IZ))/(Q5*R6)+
     1  (TS*D/2+XT*TG-Y(IZ))/(Q5*R5)-(TS*S+XT*TG+Y(IZ))/(Q6*R61)+
     1  (TS*D/2+XT*TG+Y(IZ))/(Q6*R51))+VNS(IZ,IX)
C     THESE WERE THE WING LINE SOURCE TERMS

      VNS(IZ,IX)=QT*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*ST+XLT*TG-Y(IZ))/
     1  (Q3T*R4T)-(TS*D/2+XLT*TG-Y(IZ))/(Q3T*R3T)+(TS*ST+XLT*TG+Y(IZ))/
     1  (Q4T*R41T)-(TS*D/2+XLT*TG+Y(IZ))/(Q4T*R31T)-(TS*ST*XTT*TG-
     1  Y(IZ))/(Q5T*R6T)+(TS*D/2+XTT*TG-Y(IZ))/(Q5T*R5T)-
     1  (TS*ST+XTT*TG+Y(IZ))/(Q6T*R61T)+(TS*D/2+XTT*TG+Y(IZ))/
     1  (Q6T*R51T))+VNS(IZ,IX)
C     THESE WERE THE TAIL LINE SOURCE TERMS

      VNA(IZ,IX)=-D*D*ALPHA/8*((YY-ZZ)/(YY+ZZ)**2*(1-XLS/SQRT(R7))
     1 +ZZ*XLS/(YY+ZZ)/(R7*SQRT(R7))-0.61*((YY-ZZ)/(YY+ZZ)**2*
     1 (1-XTS/SQRT(R8))+ZZ*XTS/(YY+ZZ)/(R8*SQRT(R8))))
C     THESE ARE THE DOUBLET(BODY LIFT) TERMS


      TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
      TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

      VNA(IZ,IX)=VNA(IZ,IX)-CIRC/4/PI*(F*TT+F1*TO+(Y(IZ)-SV)/
     1  (ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)-(Y(IZ)+SV)/(ZZ+
     1   (Y(IZ)+SV)**2)*(1-(XV+SV*TG)/R21))
C     THE ABOVE ARE THE WING LINE VORTEX TERMS

      TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
      TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

      VNA(IZ,IX)=VNA(IZ,IX)-CIRCT/4/PI*(FT*TTT+F1T*TOT+(Y(IZ)-SVT)/
```

```
        1          (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)-(Y(IZ)+SVT)/
        1          (ZZ+(Y(IZ)+SVT)**2)*(1-(XVT+SVT*TG)/R21T))
C                THE ABOVE ARE THE TAIL VORTEX TERMS

                 VN(IZ,IX)=+VNA(IZ,IX)+VNS(IZ,IX)

                 END DO

                 DO IZ=NY2+1,NY2+NZ    !COMPLETE SIDEWALL
                   YY=Y(IZ)**2
                   ZZ=(Z(IZ)+ALPHA*RM)**2
                   F=(XV+Y(IZ)*TG)
                   F1=(XV-Y(IZ)*TG)
                   FT=(XVT+Y(IZ)*TG)
                   FT1=(XVT-Y(IZ)*TG)

                   Q1=4*(F*F+TS*ZZ)
                   Q2=4*(F1*F1+TS*ZZ)
                   Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
                   Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
                   Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
                   Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
                   Q1T=4*(FT*FT+TS*ZZ)
                   Q2T=4*(F1T*F1T+TS*ZZ)
                   Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
                   Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
                   Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
                   Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
                   R1=SQRT(XV**2+ZZ+Y(IZ)**2)
                   R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
                   R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
                   R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
                   R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
                   R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
                   R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
                   R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
                   R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
                   R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
                   R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
                   R7=ZZ+YY+XLS*XLS
                   R8=ZZ+YY+XTS*XTS

                   R1T=SQRT(XVT**2+ZZ+YY)
                   R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
                   R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
                   R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
                   R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
                   R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
                   R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
                   R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
                   R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
                   R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
                   R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)

                 VNS(IZ,IX)=-D*D*DDD/16/BETA*Y(IZ)*(1/(R7*SQRT(R7))-
        1          0.61/(R8*SQRT(R8)))
C                THESE ARE THE BODY SOURCE TERMS ON THE WALL.

                 VNS(IZ,IX)=VNS(IZ,IX)-Q/PI/BETA*(((XL+Y(IZ)*TG)*(XL+S*TG)+ZZ)/
        1          Q3/R4-((XL+Y(IZ)*TG)*(XL+D/2*TG)+ZZ)/Q3/R3-((XL-Y(IZ)*TG)*
```

```
     1         (XL+S*TG)+ZZ)/Q4/R41+((XL-Y(IZ)*TG)*(XL+D/2*TG)+ZZ)/
     1         Q4/R31-((XT+Y(IZ)*TG)*(XT+S*TG)+ZZ)/Q5/R6+((XT+Y(IZ)*TG)*
     1         (XT+D/2*TG)+ZZ)/Q5/R5+((XT-Y(IZ)*TG)*(XT+S*TG)+ZZ)/Q6/R61
     1         -((XT-Y(IZ)*TG)*(XT+D/2*TG)+ZZ)/Q6/R51)
C            THE ABOVE ARE THE WING THICKNESS (LINE SOURCE) TERMS

             VNS(IZ,IX)=VNS(IZ,IX)-QT/PI/BETA*(((XLT+Y(IZ)*TG)*(XLT+ST*TG)
     1         +ZZ)/Q3T/R4T-((XLT+Y(IZ)*TG)*(XLT+D/2*TG)+ZZ)/Q3T/R3T-
     1         ((XLT-Y(IZ)*TG)*(XLT+ST*TG)+ZZ)/Q4T/R41T+((XLT-Y(IZ)*TG)*
     1         (XLT+D/2*TG)+ZZ)/Q4T/R31T-((XTT+Y(IZ)*TG)*(XTT+ST*TG)+ZZ)/
     1         Q5T/R6T+((XTT+Y(IZ)*TG)*(XTT+D/2*TG)+ZZ)/Q5T/R5T+((XTT-
     1         Y(IZ)*TG)*(XTT+ST*TG)+ZZ)/Q6T/R61T-((XTT-Y(IZ)*TG)*(XTT+
     1         D/2*TG)+ZZ)/Q6T/R51T)
C            THESE ARE THE TAIL THICKNESS TERMS

             VNA(IZ,IX)=-D*D*ALPHA/8*Y(IZ)*(Z(IZ)+ALPHA*RM)/(YY+ZZ)*((1-XLS/
     1        SQRT(R7))*2/(YY+ZZ)-XLS/R7/SQRT(R7)-0.61*((1-XTS/SQRT(R8))*2/
     1        (YY+ZZ)-XTS/R8/SQRT(R8)))
C            THE ABOVE ARE THE DOUBLET TERMS(BODY LIFT)

             TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
             TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

             TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
             TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

             VNA(IZ,IX)=VNA(IZ,IX)+CIRC/4/PI*(Z(IZ)+ALPHA*RM)*(TG*(TT-TO)
     1        -1/(ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)+1/(ZZ+(Y(IZ)+SV)**2)
     1        *(1-(XV+SV*TG)/R21))
C            THESE ARE THE TERMS FROM THE SWEPT LINE VORTEX

             VNA(IZ,IX)=+CIRCT/4/PI*(Z(IZ)+ALPHA*RM)*(TG*(TTT-TOT)-1/
     1         (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)+1/(ZZ+(Y(IZ)+SVT)
     1         **2)*(1-(XVT+SVT*TG)/R21T))+VNA(IZ,IX)
C            THESE ARE FOR THE TAIL VORTEX SYSTEM

             VN(IZ,IX)=VNA(IZ,IX)+VNS(IZ,IX)

             END DO                 !COMPLETE TUNNEL IS NOW DONE
             END DO


             DO IX=-20,0


C            STRETCHED DISTANCES FROM JACK POINT TO INTERSECTION OF C/L AND:
               XV=(XRVC-XJ(IX))/BETA          ! WING VORTEX
               XL=(XRVC-C/4-XJ(IX))/BETA      ! WING LEADING EDGE
               XT=(XRVC+3*C/4-XJ(IX))/BETA    ! WING TRAILING EDGE
               XLS=(XSL-XJ(IX))/BETA          ! BODY SOURCE
               XTS=(XST-XJ(IX))/BETA          ! BODY SINK(AFT)
               XVT=(XRVTC-XJ(IX))/BETA        ! TAIL VORTEX
               XLT=XVT-CT/4/BETA            ! TAIL LEADING EDGE
               XTT=XVT+3*CT/4/BETA          ! TAIL TRAILING EDGE

C            RM IS THE ARM LENGTH FROM THE POINT OF ROTATION OF
C            THE BALANCE/STING SYSTEM TO THE CP ORIGIN OF COORDS.
C            THE MODEL IS DISPLACED UPWARDS BY A DISTANCE ALPHA*RM

             RM=0.501269 !METERS
```

```
      DO IZ=1,NY2
        F=(XV+Y(IZ)*TG)
        F1=(XV-Y(IZ)*TG)
        FT=((VT+Y(IZ)*TG)
        FT1=(XVT-Y(IZ)*TG)
        YY=Y(IZ)**2
        ZZ=(Z(IZ)+ALPHA*RM)**2
        Q1=4*(F*F+TS*ZZ)
        Q2=4*(F1*F1+TS*ZZ)
        Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
        Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
        Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
        Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
        Q1T=4*(FT*FT+TS*ZZ)
        Q2T=4*(F1T*F1T+TS*ZZ)
        Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
        Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
        Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
        Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
        R1=SQRT(XV**2+ZZ+YY)
        R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
        R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
        R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
        R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
        R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
        R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
        R7=ZZ+YY+XLS*XLS
        R8=ZZ+YY+XTS*XTS

        R1T=SQRT(XVT**2+ZZ+YY)
        R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
        R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
        R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
        R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
        R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
        R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
        R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
        R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)


      VNS(IZ,IX)=-D*D*DDD*(Z(IZ)+ALPHA*RM)/16/BETA*(1/(R7*SQRT(R7))
     1  -0.61/(R8*SQRT(R8)))
C     THESE ABOVE ARE THE BODY SOURCE TERMS
      VNS(IZ,IX)=-Q*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*S+XL*TG-Y(IZ))/(Q3*R4)-
     1  (TS*D/2+XL*TG-Y(IZ))/(Q3*R3)+(TS*S+XL*TG+Y(IZ))/(Q4*R41)-
     1  (TS*D/2+XL*TG+Y(IZ))/(Q4*R31)-(TS*S+XT*TG-Y(IZ))/(Q5*R6)+
     1  (TS*D/2+XT*TG-Y(IZ))/(Q5*R5)-(TS*S+XT*TG+Y(IZ))/(Q6*R61)+
     1  (TS*D/2+XT*TG+Y(IZ))/(Q6*R51))+VNS(IZ,IX)
C     THESE WERE THE WING LINE SOURCE TERMS

      VNS(IZ,IX)=-QT*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*ST+XLT*TG-Y(IZ))/
     1  (Q3T*R4T)-(TS*D/2+XLT*TG-Y(IZ))/(Q3T*R3T)+(TS*ST+XLT*TG+Y(IZ))/
     1  (Q4T*R41T)-(TS*D/2+XLT*TG+Y(IZ))/(Q4T*R31T)-(TS*ST*XTT*TG-
```

```
     1      Y(IZ))/(Q5T*R6T)+(TS*D/2+XTT*TG-Y(IZ))/(Q5T*R5T)-
     1      (TS*ST+XTT*TG+Y(IZ))/(Q6T*R61T)+(TS*D/2+XTT*TG+Y(IZ))/
     1      (Q6T*R51T))+VNS(IZ,IX)
C         THESE WERE THE TAIL LINE SOURCE TERMS

          VNA(IZ,IX)=D*D*ALPHA/8*((YY-ZZ)/(YY+ZZ)**2*(1-XLS/SQRT(R7))
     1     +ZZ*XLS/(YY+ZZ)/(R7*SQRT(R7))-0.61*((YY-ZZ)/(YY+ZZ)**2*
     1     (1-XTS/SQRT(R8))+ZZ*XTS/(YY+ZZ)/(R8*SQRT(R8))))
C         THESE ARE THE DOUBLET(BODY LIFT) TERMS


          TT-4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
          TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

          VNA(IZ,IX)=VNA(IZ,IX)+CIRC/4/PI*(F*TT+F1*TO+(Y(IZ)-SV)/
     1      (ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)-(Y(IZ)+SV)/(ZZ+
     1       (Y(IZ)+SV)**2)*(1-(XV+SV*TG)/R21))
C         THE ABOVE ARE THE WING LINE VORTEX TERMS

          TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
          TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

          VNA(IZ,IX)=VNA(IZ,IX)+CIRCT/4/PI*(FT*TTT+F1T*TOT+(Y(IZ)-SVT)/
     1      (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)-(Y(IZ)+SVT)/
     1      (ZZ+(Y(IZ)+SVT)**2)*(1-(XVT+SVT*TG)/R21T))
C         THE ABOVE ARE THE TAIL VORTEX TERMS

          VNJ(IZ,IX)=VNA(IZ,IX)+VNS(IZ,IX)

          END DO

          DO IZ=NY2+NZ+1,NY+NZ    !UPPER WALL
            F=(XV+Y(IZ)*TG)
            F1=(XV-Y(IZ)*TG)
            FT=(XVT+Y(IZ)*TG)
            FT1=(XVT-Y(IZ)*TG)
            YY=Y(IZ)**2
            ZZ=(Z(IZ)+ALPHA*RM)**2
            Q1=4*(F*F+TS*ZZ)
            Q2=4*(F1*F1+TS*ZZ)
            Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
            Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
            Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
            Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
            Q1T=4*(FT*FT+TS*ZZ)
            Q2T=4*(F1T*F1T+TS*ZZ)
            Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
            Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
            Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
            Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
            R1=SQRT(XV**2+ZZ+YY)
            R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
            R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
            R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
            R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
            R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
```

```
      R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
      R7=ZZ+YY+XLS*XLS
      R8=ZZ+YY+XTS*XTS

      R1T=SQRT(XVT**2+ZZ+YY)
      R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
      R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
      R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
      R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
      R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
      R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
      R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
      R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
      R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
      R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)


      VNS(IZ,IX)=+D*D*DDD*(Z(IZ)+ALPHA*RM)/16/BETA*(1/(R7*SQRT(R7))
     1  -0.61/(R8*SQRT(R8)))
C     THESE ABOVE ARE THE BODY SOURCE TERMS
      VNS(IZ,IX)=Q*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*S+XL*TG-Y(IZ))/(Q3*R4)-
     1  (TS*D/2+XL*TG-Y(IZ))/(Q3*R3)+(TS*S+XL*TG+Y(IZ))/(Q4*R41)-
     1  (TS*D/2+XL*TG+Y(IZ))/(Q4*R31)-(TS*S+XT*TG-Y(IZ))/(Q5*R6)+
     1  (TS*D/2+XT*TG-Y(IZ))/(Q5*R5)-(TS*S+XT*TG+Y(IZ))/(Q6*R61)+
     1  (TS*D/2+XT*TG+Y(IZ))/(Q6*R51))+VNS(IZ,IX)
C     THESE WERE THE WING LINE SOURCE TERMS

      VNS(IZ,IX)=QT*(Z(IZ)+ALPHA*RM)/PI/BETA*((TS*ST+XLT*TG-Y(IZ))/
     1  (Q3T*R4T)-(TS*D/2+XLT*TG-Y(IZ))/(Q3T*R3T)+(TS*ST+XLT*TG+Y(IZ))/
     1  (Q4T*R41T)-(TS*D/2+XLT*TG+Y(IZ))/(Q4T*R31T)-(TS*ST*XTT*TG-
     1  Y(IZ))/(Q5T*R6T)+(TS*D/2+XTT*TG-Y(IZ))/(Q5T*R5T)-
     1  (TS*ST+XTT*TG+Y(IZ))/(Q6T*R61T)+(TS*D/2+XTT*TG+Y(IZ))/
     1  (Q6T*R51T))+VNS(IZ,IX)
C     THESE WERE THE TAIL LINE SOURCE TERMS

      VNA(IZ,IX)=-D*D*ALPHA/8*((YY-ZZ)/(YY+ZZ)**2*(1-XLS/SQRT(R7))
     1 +ZZ*XLS/(YY+ZZ)/(R7*SQRT(R7))-0.61*((YY-ZZ)/(YY+ZZ)**2*
     1 (1-XTS/SQRT(R8))+ZZ*XTS/(YY+ZZ)/(R8*SQRT(R8))))
C     THESE ARE THE DOUBLET(BODY LIFT) TERMS


      TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
      TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

      VNA(IZ,IX)=VNA(IZ,IX)-CIRC/4/PI*(F*TT+F1*TO+(Y(IZ)-SV)/
     1  (ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)-(Y(IZ)+SV)/(ZZ+
     1   (Y(IZ)+SV)**2)*(1-(XV+SV*TG)/R21))
C     THE ABOVE ARE THE WING LINE VORTEX TERMS

      TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
      TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

      VNA(IZ,IX)=VNA(IZ,IX)-CIRCT/4/PI*(FT*TTT+F1T*TOT+(Y(IZ)-SVT)/
     1  (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)-(Y(IZ)+SVT)/
     1  (ZZ+(Y(IZ)+SVT)**2)*(1-(XVT+SVT*TG)/R21T))
C     THE ABOVE ARE THE TAIL VORTEX TERMS

      VNJ(IZ,IX)=+VNA(IZ,IX)+VNS(IZ,IX)

      END DO
```

```
          DO IZ=NY2+1,NY2+NZ   !COMPLETE SIDEWALL
            YY=Y(IZ)**2
            ZZ=(Z(IZ)+ALPHA*RM)**2
            F=(XV+Y(IZ)*TG)
            F1=(XV-Y(IZ)*TG)
            FT=(XVT+Y(IZ)*TG)
            FT1=(XVT-Y(IZ)*TG)

            Q1=4*(F*F+TS*ZZ)
            Q2=4*(F1*F1+TS*ZZ)
            Q3=4*((XL+Y(IZ)*TG)**2+TS*ZZ)
            Q4=4*((XL-Y(IZ)*TG)**2+TS*ZZ)
            Q5=4*((XT+Y(IZ)*TG)**2+TS*ZZ)
            Q6=4*((XT-Y(IZ)*TG)**2+TS*ZZ)
            Q1T=4*(FT*FT+TS*ZZ)
            Q2T=4*(F1T*F1T+TS*ZZ)
            Q3T=4*((XLT+Y(IZ)*TG)**2+TS*ZZ)
            Q4T=4*((XLT-Y(IZ)*TG)**2+TS*ZZ)
            Q5T=4*((XTT+Y(IZ)*TG)**2+TS*ZZ)
            Q6T=4*((XTT-Y(IZ)*TG)**2+TS*ZZ)
            R1=SQRT(XV**2+ZZ+Y(IZ)**2)
            R2=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)-SV)**2)
            R21=SQRT((XV+TG*SV)**2+ZZ+(Y(IZ)+SV)**2)
            R3=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R31=SQRT((XL+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R4=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)-S)**2)
            R41=SQRT((XL+TG*S)**2+ZZ+(Y(IZ)+S)**2)
            R5=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R51=SQRT((XT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R6=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)-S)**2)
            R61=SQRT((XT+TG*S)**2+ZZ+(Y(IZ)+S)**2)
            R7=ZZ+YY+XLS*XLS
            R8=ZZ+YY+XTS*XTS

            R1T=SQRT(XVT**2+ZZ+YY)
            R2T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)-SVT)**2)
            R21T=SQRT((XVT+TG*SVT)**2+ZZ+(Y(IZ)+SVT)**2)
            R3T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R31T=SQRT((XLT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R4T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
            R41T=SQRT((XLT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)
            R5T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)-D/2)**2)
            R51T=SQRT((XTT+TG*D/2)**2+ZZ+(Y(IZ)+D/2)**2)
            R6T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)-ST)**2)
            R61T=SQRT((XTT+TG*ST)**2+ZZ+(Y(IZ)+ST)**2)

          VNS(IZ,IX)=-D*D*DDD/16/BETA*Y(IZ)*(1/(R7*SQRT(R7))-
     1      0.61/(R8*SQRT(R8)))
C         THESE ARE THE BODY SOURCE TERMS ON THE WALL.

          VNS(IZ,IX)=VNS(IZ,IX)-Q/PI/BETA*(((XL+Y(IZ)*TG)*(XL+S*TG)+ZZ)/
     1      Q3/R4-((XL+Y(IZ)*TG)*(XL+D/2*TG)+ZZ)/Q3/R3-((XL-Y(IZ)*TG)*
     1      (XL+S*TG)+ZZ)/Q4/R41+((XL-Y(IZ)*TG)*(XL+D/2*TG)+ZZ)/
     1      Q4/R31-((XT+Y(IZ)*TG)*(XT+S*TG)+ZZ)/Q5/R6+((XT+Y(IZ)*TG)*
     1      (XT+D/2*TG)+ZZ)/Q5/R5+((XT-Y(IZ)*TG)*(XT+S*TG)+ZZ)/Q6/R61
     1      -((XT-Y(IZ)*TG)*(XT+D/2*TG)+ZZ)/Q6/R51)
C         THE ABOVE ARE THE WING THICKNESS (LINE SOURCE) TERMS

          VNS(IZ,IX)=VNS(IZ,IX)-QT/PI/BETA*(((XLT+Y(IZ)*TG)*(XLT+ST*TG)
```

```
    1     +ZZ)/Q3T/R4T-((XLT+Y(IZ)*TG)*(XLT+D/2*TG)+ZZ)/Q3T/R3T-
    1     ((XLT-Y(IZ)*TG)*(XLT+ST*TG)+ZZ)/Q4T/R41T+((XLT-Y(IZ)*TG)*
    1     (XLT+D/2*TG)+ZZ)/Q4T/R31T-((XTT+Y(IZ)*TG)*(XTT+ST*TG)+ZZ)/
    1     Q5T/R6T+((XTT+Y(IZ)*TG)*(XTT+D/2*TG)+ZZ)/Q5T/R5T+((XTT-
    1     Y(IZ)*TG)*(XTT+ST*TG)+ZZ)/Q6T/R61T-((XTT-Y(IZ)*TG)*(XTT+
    1     D/2*TG)+ZZ)/Q6T/R51T)
C        THESE ARE THE TAIL THICKNESS TERMS

         VNA(IZ,IX)=-D*D*ALPHA/8*Y(IZ)*(Z(IZ)+ALPHA*RM)/(YY+ZZ)*((1-XLS/
    1     SQRT(R7))*2/(YY+ZZ)-XLS/R7/SQRT(R7)-0.61*((1-XTS/SQRT(R8))*2/
    1     (YY+ZZ)-XTS/R8/SQRT(R8)))
C        THE ABOVE ARE THE DOUBLET TERMS(BODY LIFT)

         TT=4/Q1*((TS*SV+XV*TG-Y(IZ))/R2-(XV*TG-Y(IZ))/R1)
         TO=4/Q2*((TS*SV+XV*TG+Y(IZ))/R21-(XV*TG+Y(IZ))/R1)

         TTT=4/Q1T*((TS*SVT+XVT*TG-Y(IZ))/R2T-(XVT*TG-Y(IZ))/R1T)
         TOT=4/Q2T*((TS*SVT+XVT*TG+Y(IZ))/R21T-(XVT*TG+Y(IZ))/R1T)

         VNA(IZ,IX)=VNA(IZ,IX)+CIRC/4/PI*(Z(IZ)+ALPHA*RM)*(TG*(TT-TO)
    1     -1/(ZZ+(Y(IZ)-SV)**2)*(1-(XV+SV*TG)/R2)+1/(ZZ+(Y(IZ)+SV)**2)
    1     *(1-(XV+SV*TG)/R21))
C        THESE ARE THE TERMS FROM THE SWEPT LINE VORTEX

         VNA(IZ,IX)=+CIRCT/4/PI*(Z(IZ)+ALPHA*RM)*(TG*(TTT-TOT)-1/
    1     (ZZ+(Y(IZ)-SVT)**2)*(1-(XVT+SVT*TG)/R2T)+1/(ZZ+(Y(IZ)+SVT)
    1     **2)*(1-(XVT+SVT*TG)/R21T))+VNA(IZ,IX)
C        THESE ARE FOR THE TAIL VORTEX SYSTEM

         VNJ(IZ,IX)=VNA(IZ,IX)+VNS(IZ,IX)

         END DO                   !COMPLETE TUNNEL IS NOW DONE
         END DO

C        WE NOW WRITE A FILE FOR VNO TO FEED JACK_DISPL!
C        FIRST WE MUST DEFINE A NEW X-WISE COORDINATE MEASURED
C        FROM THE HALF INCH POINT ON THE WALL PLATES.

         DO IX=-NX,NX
           X_LOC(IX)=X(IX)+(27.5*0.0254) !CP IS LOCATED AT 28 INCH POINT
         END DO

         OPEN(UNIT=8,NAME='VNO.DAT',STATUS='NEW')
         WRITE(8,40) CL,  MACH
40       FORMAT('THIS DATA IS FOR CL='F4.2,3X,'MACH=',F4.2)
         WRITE(8,*)  1.79959,  0.3302
         WRITE(8,42)  1.0
42       FORMAT('FREESTREAM VELOCITY = ',F6.1)
         WRITE(8,43)
43       FORMAT('UNIT OF MEASUREMENT IS METERS')
         WRITE(8,44)  (2*NX+1)
44       FORMAT('NUMBER OF X LOCATIONS = ',I5)
         WRITE(8,45)
45       FORMAT('LOCATIONS')
         WRITE(8,*) (X_LOC(IX),IX=-NX,NX)
         WRITE(8,46)  NY
46       FORMAT('NUMBER OF PANELS IN Y DIRECTION = ',I3)
         WRITE(8,47) (A/NY)
47       FORMAT('WIDTH OF PANELS = ',F6.5)
         WRITE(8,48)
```

```
48          FORMAT('LOCATIONS:')
            WRITE(8,*) (Y(IZ),IZ=1,NY2)
            WRITE(8,49)  NZ
49          FORMAT('NUMBER OF PANELS IN Z DIRECTION = ',I3)
            WRITE(8,50)  (A/NZ)
50          FORMAT('WIDTH OF PANELS = ',F6.5)
            WRITE(8,51)
51          FORMAT(2X,//////)

            DO IX=-NX,NX
            WRITE(8,*)  (VN(IZ,IX),IZ=1,NY+NZ)
            END DO
            CLOSE(8)

C           WE NOW WRITE A FILE FOR VNJ TO FEED JACK_DISPL!
C           REDEFINE THE JACK LOCATIONS NOW FROM THE 1/2 INCH STATION

            XJ(-20)=4.25*.0254
            XJ(-19)=10.*.0254
            XJ(-18)=15.*.0254
            XJ(-17)=19.*.0254
            XJ(-16)=22.*.0254
            XJ(-15)=24.*.0254
            XJ(-14)=25.5*.0254
            XJ(-13)=27.*.0254
            XJ(-12)=28.5*.0254
            XJ(-11)=30.*.0254
            XJ(-10)=31.5*.0254
            XJ(-9)=33.*.0254
            XJ(-8)=35.*.0254
            XJ(-7)=37.*.0254
            XJ(-6)=39.*.0254
            XJ(-5)=42.*.0254
            XJ(-4)=46.*.0254
            XJ(-3)=51.*.0254
            XJ(-2)=56.*.0254
            XJ(-1)=61.*.0254
            XJ(0)=66.5*.0254

            OPEN(UNIT=7,NAME='VNJ.DAT',STATUS='NEW')
            WRITE(7,61)  17
61          FORMAT('NUMBER OF JACKS =',I3)
            WRITE(7,62)
62          FORMAT('DISTANCES IN METERS')
            WRITE(7,*) (XJ(IX),IX=-20,-4)
            WRITE(7,63)
63          FORMAT('NUMBER OF Y PANELS'5X,'NUMBER OF Z PANELS')
            WRITE(7,*)  NY,  NZ
            WRITE(7,64)
64          FORMAT(2X,/////)

            DO IX=-20,-4
            WRITE(7,*) (VNJ(IZ,IX),IZ=1,NY+NZ)
            END DO
            CLOSE(7)

            END
```

APPENDIX E - VEEXPHINO FORTRAN LISTING

Primary Symbols

A	Tunnel height
A0, AX	Normal velocities at panel centers due to horseshoe vortices on a panel in a ring at IX = -NX, fore and aft symmetric and unsymmetric contributions
D	Tunnel breadth
DYDZ	Ratio of floor panel width to wall panel height
G0	Axial disturbance velocities at panel center due to a Green's source panel in a ring at IX = -NX
MACH	Mach number
MEASVN, MEASVX	Measured wall values of normal and axial disturbance velocities
PHINO	Computed wall-free normal velocities at panel centers
VX	Calculated axial velocity increment due to measured normal velocities (see report)
VN	Residual normal velocity field to be nulled by the walls. Input into PHIXZM provides flow distortion field at model

```
C                  PROGRAM VEEX-PHINO

C            THIS PROGRAM COMPUTES A FIRST ORDER APPROXIMATION TO
C            WALL-FREE NORMAL VELOCITIES AT RECTANGULAR TUNNEL WALLS
C            PRODUCED BY A TEST MODEL. INPUTS REQUIRED ARE MEASURED
C            NORMAL AND AXIAL DISTURBANCE VELOCITIES AT THE WALLS OR
C            RECTANGULAR CONTROL SURFACES DURING A TEST.
C            THE FIRST PART OF THE COMPUTATION GETS THE AXIAL DISTURBANCE
C            VELOCITY FIELD AT THE WALLS CAUSED BY THE MEASURED TEST
C            NORMAL VELOCITIES AT THE WALLS.

C            A RECTANGULAR TUNNEL IS ASSUMED OF HEIGHT=A AND
C            WIDTH=D

C            SELECT AN EVEN!!  NUMBER OF PANELS ON THE VERTICAL
C            WALL(NZ). THIS FIXES THE HEIGHT OF THE PANELS WHICH IS
C            ALSO TAKEN AS THE LENGTH OF THE PANELS. THE LENGTH OF
C            TUNNEL CONSIDERED CAN BE SET BY SELECTING THE NUMBER OF
C            PANELS FOREWARD AND AFT OF THE ORIGIN,NX.

C            PRANDTL-GLAUERT COMPRESSIBILTY CORRECTIONS ARE USED

C            A PARAMETER 'M' IS USED TO FIX THE NUMBER OF SOURCE
C            IMAGES USED BEFORE SMEARING TO OBTAIN A CLOSED FORMULA
C            FOR THE REMAINING SOURCES OUT TO INFINITY.


             DIMENSION      X(-80:80), Y(80),Z(80),
     1       A0(10,80),YDM(5,80),YDP(5,80),YSM(5,80),YSP(5,80),YD(5),
     1       YP(5),YD6(16:80),YS6(16:80),ZD(6:15),ZDP2(6:10,6:15),
     1       ZDM2(6:10,6:15),ZDP(6:10),ZDM(6:10),YZ1(5,6:15),YZ2(5,6:15),
     1       YZ3(5,6:15),YZ4(5,6:15),ZY1(6:10,5),ZY2(6:10,5),ZY3(6:10,5),
     1       ZY4(6:10,5),ZDP3(6:10,6:15),ZDM3(6:10,6:15),YD1(5,16:80),
     1       YD2(5,16:80),YD3(5,16:80),YD4(5,16:80),YR1(6:10,16:80),
     1       YR2(6:10,16:80),YR3(6:10,16:80),YR4(6:10,16:80),
     1       AX(10,80,-80:80),PHINO(80,-80:80),VX(80,-80:80),
     1       MEASVN(80,-80:80),MEASVX(80,-80:80),VN(80,-80:80),W(80,-80:80),
     1       PT(20:20,40,80),PJ(-20:20,40,80),G0(40,80,-80:80),
     1       Q(0:20,40,80), R(0:20,40,80),S(0:20,40,80),T(0:20,40,80),
     1       YPP(40,80),YDD(40,80)

             PARAMETER (PI=3.14159)
             REAL MACH
             TYPE 10
10           FORMAT(10X,'ENTER NX  NZ   MACH  A  D AND M')
             READ(5,*) NX, NZ, MACH, A, D,M
             BETA=SQRT(1-MACH*MACH)
             NZ2=NZ/2
             NY2=NINT(D*NZ2/A+0.1)
             NY=NY2*2

C            NOW SET UP THE COORDINATES OF THE PANELS.

             DO IX=-NX,NX
               X(IX)=IX*A/NZ/BETA
             END DO
             DO IZ=1,NY2
               Y(IZ)=(IZ-0.5)*D/NY
               Z(IZ)=A/2
```

```
      END DO
      DO IZ=(NY2+1),(NY2+NZ)
        Y(IZ)=D/2
        Z(IZ)=((NZ+NY+1)/2.00-IZ)*A/NZ
      END DO
      DO IZ=(NZ+NY2+1),(NZ+NY)
        Y(IZ)=((NZ+NY+0.5)-IZ)*D/NY
        Z(IZ)=-A/2
      END DO

      DYDZ=D/A*NZ/NY   !RATIO OF PANEL FLOOR AND WALL WIDTHS

C     SET UP COMMON REPEATING TERMS TO SAVE TIME!

      DO IZ=1,NY+NZ
      DO JZ=1,NZ2+NY2
        YDD(JZ,IZ)=Y(JZ)-Y(IZ)
        YPP(JZ,IZ)=Y(JZ)+Y(IZ)
      DO I=-M,M
        PT(I,JZ,IZ)=(2*A*I+Z(JZ)-Z(IZ))**2
        PJ(I,JZ,IZ)=(2*A*I+A-Z(JZ)-Z(IZ))**2
      END DO
      DO L=0,M
        DL=D*L
        DP=D*(L+1)
        Q(L,JZ,IZ)=(DL+YDD(JZ,IZ))**2
        R(L,JZ,IZ)=(DP-YPP(JZ,IZ))**2
        S(L,JZ,IZ)=(DL+YPP(JZ,IZ))**2
        T(L,JZ,IZ)=(DP-YDD(JZ,IZ))**2
      END DO
      END DO
      END DO

C     THE PRIMARY COMPUTATION BEGINS HERE!
C     THE FIRST STEP IS TO GET THE PHI-X VALUES FROM A RING OF
C     SOURCE PANELS AT IX=-NX

      DO KX=-NX+1,NX
        XX=X(-NX)-X(KX)
        XS=XX**2

C     FOR TIME SAVING WE SMEAR THE SOURCE IMAGES BEYOND M
C     PAIRS IN BOTH HORIZONTAL AND VERTICAL DIRECTIONS.

      E=D/A* NZ*NZ*PI*BETA
      EEE=1/E
      DM=(M+1)*D
      DMS=DM**2
      HM=(M+0.75)*2*A
      HM2=(M+0.25)*2*A
      HMS=HM**2
      HM2S=HM2**2
      GS=EEE/BETA*(ATAN(-XX*SQRT((XS+DMS+HMS)/(HM*DM)))+
     1         ATAN(-XX*SQRT((XS+DMS+HM2S)/(HM2*DM))))
C     THESE ARE THE PHI-X VALUES FROM THE SMEARED SOURCES

      DO IZ=1,NY+NZ
      DO JZ=1,NY2       ! JUST THE FLOOR SOURCES
        DO I=-M,M
         P1=PT(I,JZ,IZ)+XS
```

```
                 P2=PJ(I,JZ,IZ)+XS
                DO L=0,M
                 TT1=P1+Q(L,JZ,IZ)
                 TT1=TT1*SQRT(TT1)
                 TT2=P2+R(L,JZ,IZ)
                 TT2=TT2*SQRT(TT2)
                 TT3=P1+S(L,JZ,IZ)
                 TT3=TT3*SQRT(TT3)
                 TT4=P2+T(L,JZ,IZ)
                 TT4=TT4*SQRT(TT4)

              QQ=A*A/(2*PI*BETA*NZ*NZ)
              TTT=(1/TT1+1/TT2+1/TT3+1/TT4)*QQ

              GO(JZ,IZ,KX)=-XX*TTT/BETA+GO(JZ,IZ,KX)

              END DO
              END DO

C             THE TERM DYDZ APPEARS BELOW TO ACCOUNT FOR THE DIFFER-
C             ENCE IN PANEL WIDTHS OF FLOUR AND WALL,IF ANY.

              GO(JZ,IZ,KX)=(GO(JZ,IZ,KX)+GS)*DYDZ
              END DO

              DO JZ=NY2+1,NZ2      !LOWER HALF OF THE WALL SOURCES

              DO I=-M,M
                P1=PT(I,JZ,IZ)+XS
                P2=PJ(I,JZ,IZ)+XS
              DO L=0,M
                 TT1=P1+Q(L,JZ,IZ)
                 TT1=TT1*SQRT(TT1)
                 TT2=P2+R(L,JZ,IZ)
                 TT2=TT2*SQRT(TT2)
                 TT3=P1+S(L,JZ,IZ)
                 TT3=TT3*SQRT(TT3)
                 TT4=P2+T(L,JZ,IZ)
                 TT4=TT4*SQRT(TT4)

              TTT=(1/TT1+1/TT2+1/TT3+1/TT4)*QQ

              GO(JZ,IZ,KX)=-XX*TTT/BETA+GO(JZ,IZ,KX)

              END DO
              END DO
              GO(JZ,IZ,KX)=GO(JZ,IZ,KX)+GS
              END DO
              END DO
              END DO

C             NOTE THAT THE FUNCTION GO IS USED BELOW TO CREATE
C             Gi DERIVED IN THE REPORT. USE OF THE REPEATING GO SAVES TIME
C             AND MEMORY SPACE WHEN A LARGE NUMBER OF PANELS ARE USED.

C             THIS COMPLETES THE WORK FOR ALL THE SOURCES LOCATED AT JX
C             EQUAL TO -NX  IN THE LOWER HALF TUNNEL

              OPEN(UNIT =1,NAME='MEASVX.DAT',STATUS='OLD')
              READ(1,*) ((MEASVX(IZ,IX), IZ=1,NY+NZ),IX=-NX,NX)
```

```
        CLOSE(1)
        OPEN(UNIT=2,NAME='MEASVN.DAT',STATUS='OLD')
        READ(2,*)  ((MEASVN(IZ,IX),IZ=1,NY+NZ),IX=-NX,NX)
        CLOSE(2)

        DO IX=-NX,NX
        DO IZ=1,NY+NZ
          LZ=NY+NZ+1-IZ
        DO JX=-NX,IX-1
          KX=-NX-(JX-IX)
        DO JZ=1,NY2+NZ2

          VX(IZ,IX)=VX(IZ,IX)+MEASVN(JZ,JX)*GO(JZ,IZ,KX)
        END DO
        DO JZ=(NY2+NZ2+1),NY+NZ
          KZ=NY+NZ+1-JZ
          VX(IZ,IX)=VX(IZ,IX)+MEASVN(JZ,JX)*GO(KZ,LZ,KX)
        END DO
        END DO

        DO JX=(IX+1),NX
          LX=-NX+JX-IX
        DO JZ=1,NY2+NZ2
          VX(IZ,IX)=VX(IZ,IX)-MEASVN(JZ,JX)*GO(JZ,IX,LX)
        END DO
        DO JZ=NZ2+NY2+1,NZ+NY
          KZ=NY+NZ+1-JZ
        VX(IZ,IX)=VX(IZ,IX)+MEASVN(JZ,JX)*GO(KZ,LZ,KX)
        END DO
        END DO

        DO JX=(IX+1),NX
          LX=-NX+JX-IX
        DO JZ=1,NY2+NZ2
          VX(IZ,IX)=VN(IZ,IX)-MEASVN(JZ,JZ)*GO(JZ,IZ,LX)
        END DO
        DO JZ=NY2+NZ2+1,NY+NZ
          KZ=NY+NZ+1-JZ
        VX(IZ,IX)=VN(IZ,IX)-MEASVN(JZ,JX)*GO(KZ,LZ,LX)
        END DO
        END DO

C       THIS IS THE PHI-X OF THE MEASURED NORMAL VELOCITIES
C       DISTRIBUTION.  NOW SUBTRACT THIS FROM THE MEASVX VALUES
C       FOR INPUT INTO THE FOLLOWING PHINO COMPUTATION.

        VX(IZ,IX)=MEASVX(IZ,IX)-VX(IZ,IX)
        END DO
        END DO

C       NOW USING THE COMPUTED VX GET WALL-FREE NORMAL VELOCITIES

C       THIS PROGRAM COMPUTES THE RESIDUAL NORMAL VELOCITIES
C       AT THE TUNNEL WALLS GIVEN THE INPUT FUNCTION VX FROM THE
C       PRECEDING WORK. THE OUTPUT CALLED PHINWALL.DAT IS
C       THEN  USED TO COMPUTE THE INTERFERENCE VELOCITIES
C       AT THE POSITION OF THE MODEL USING PROGRAM 'PHIXZM'.
C       IT ALSO USES THE INPUT FILE CALLED MEASVN THAT
C       IS THE MEASURED WALL SLOPE PLUS BOUNDARY LAYER SLOPE.
C       IT DEVELOPS THE PANEL EQUATIONS FOR THE NORMAL VELOCITIES
```

```
C       PRODUCED BY A HORSESHOE VORTEX LYING AT THE CENTER OF A PANEL

        EE=2*A/NZ/BETA   !CIRCULATION FOR PANEL OF UNIT VX
        SA=A/2/NZ
        SFA=A/2/NZ*DYDZ
        DO IZ=1,NY2    ! FIELD POINT ON THE FLOOR
        DO JZ=1,NY2    ! VORTEX ALSO ON THE FLOOR
         YDP(JZ,IZ)=Y(IZ)-Y(JZ)+SFA
         YDM(JZ,IZ)=Y(IZ)-Y(JZ)-SFA
         YSP(JZ,IZ)=Y(IZ)+Y(JZ)+SFA

         YSM(JZ,IZ)=Y(IZ)+Y(JZ)-SFA

        AO(JZ,IZ)=EE*(1/YDP(JZ,IZ)-1/YDM(JZ,IZ)+
    1        1/YSP(JZ,IZ)-1/YSM(JZ,IZ))     !CASE ONE
        END DO
         YD(IZ)=Y(IZ)-D/2
         YP(IZ)=Y(IZ)+D/2
        DO JZ=NY2+1,NY2+NZ2  ! VORTEX ON  LOWER HALF OF WALL
         ZDP(JZ)=A/2-Z(JZ)+SA
         ZDM(JZ)=A/2-Z(JZ)-SA
         ZY1(JZ,IZ)=YD(IZ)*YD(IZ)+ZDM(JZ)*ZDM(JZ)
         ZY2(JZ,IZ)=YD(IZ)*YD(IZ)+ZDP(JZ)*ZDP(JZ)
         ZY3(JZ,IZ)=YP(IZ)*YP(IZ)+ZDM(JZ)*ZDM(JZ)
         ZY4(JZ,IZ)=YP(IZ)*YP(IZ)+ZDP(JZ)*ZDP(JZ)

            AO(JZ,IZ)=EE*(YD(IZ)/ZY1(JZ,IZ)-YD(IZ)/ZY2(JZ,IZ)-
    1          YP(IZ)/ZY3(JZ,IZ)+YP(IZ)/ZY4(JZ,IZ))   !CASE TWO
        END DO
        END DO

        DO IZ=NY2+1,NY2+NZ  ! FIELD POINT ON THE WALL

         ZD(IZ)=Z(IZ)-A/2

        DO JZ=1,NY2    !  VORTEX ON THE FLOOR

         YDP(JZ,IZ)=Y(IZ)-Y(JZ)+SF
         YDM(JZ,IZ)=Y(IZ)-Y(JZ)-SF
         YSP(JZ,IZ)=Y(IZ)+Y(JZ)+SF
         YSM(JZ,IZ)=Y(IZ)+Y(JZ)-SF
         YZ1(JZ,IZ)=ZD(IZ)*ZD(IZ)+YDP(JZ,IZ)*YDP(JZ,IZ)
         YZ2(JZ,IZ)=ZD(IZ)*ZD(IZ)+YDM(JZ,IZ)*YDM(JZ,IZ)
         YZ3(JZ,IZ)=ZD(IZ)*ZD(IZ)+YSP(JZ,IZ)*YSP(JZ,IZ)
         YZ4(JZ,IZ)=ZD(IZ)*ZD(IZ)+YSM(JZ,IZ)*YSM(JZ,IZ)

         AO(JZ,IZ)=-EE*ZD(IZ)*(1/YZ1(JZ,IZ)-1/YZ2(JZ,IZ)+
    1           1/YZ3(JZ,IZ)-1/YZ4(JZ,IZ))  !  THREE
        END DO

        DO JZ=NY2+1, NY2+NZ2  ! VORTEX ALSO ON LOWER HALF OF WALL
         ZDP2(JZ,IZ)=Z(IZ)-Z(JZ)+SA
         ZDM2(JZ,IZ)=Z(IZ)-Z(JZ)-SA
         ZDP3(JZ,IZ)=ZDP2(JZ,IZ)*ZDP2(JZ,IZ)+D*D
         ZDM3(JZ,IZ)=ZDM2(JZ,IZ)*ZDM2(JZ,IZ)+D*D

         AO(JZ,IZ)=EE*(1/ZDP2(JZ,IZ)-1/ZDM2(JZ,IZ))-EE*(ZDP2(JZ,IZ)/
    1     (ZDP2(JZ,IZ)*ZDP2(JZ,IZ)+D*D)-ZDM2(JZ,IZ)/(ZDM2(JZ,IZ)*
    1     ZDM2(JZ,IZ)+D*D))             !  FOUR
```

```
              END DO
              END DO

              DO IZ=NY2+NZ+1,NY+NZ
              DO JZ=1,NY2
                YDP(JZ,IZ)=Y(IZ)-Y(JZ)+SFA
                YDM(JZ,IZ)=Y(IZ)-Y(JZ)-SFA
                YSP(JZ,IZ)=Y(IZ)+Y(JZ)+SFA
                YSM(JZ,IZ)=Y(IZ)+Y(JZ)-SFA
                YD1(JZ,IZ)=YDP(JZ,IZ)*YDP(JZ,IZ)+A*A
                YD2(JZ,IZ)=YDM(JZ,IZ)*YDM(JZ,IZ)+A*A
                YD3(JZ,IZ)=YSP(JZ,IZ)*YSP(JZ,IZ)+A*A
                YD4(JZ,IZ)=YSM(JZ,IZ)*YSM(JZ,IZ)+A*A

                AO(JZ,IZ)=-EE*(YDP(JZ,IZ)/YD1(JZ,IZ)-
     1                        YDM(JZ,IZ)/YD2(JZ,IZ)+
     1                        YSP(JZ,IZ)/YD3(JZ,IZ)-
     1                        YSM(JZ,IZ)/YD4(JZ,IZ))
C                                                        FIVE
              END DO

              YD6(IZ)=Y(IZ)-D/2
              YS6(IZ)=Y(IZ)+D/2

              DO JZ=NY2+1,NY2+NZ2    !  VORTEX ON THE LOWER HALF WALL
                ZDP(JZ)=-A/2-Z(JZ)+SA
                ZDM(JZ)=-A/2-Z(JZ)-SA

                YR1(JZ,IZ)=YD6(IZ)*YD6(IZ)+ZDP(JZ)*ZDP(JZ)
                YR2(JZ,IZ)=YD6(IZ)*YD6(IZ)+ZDM(JZ)*ZDM(JZ)
                YR3(JZ,IZ)=YS6(IZ)*YS6(IZ)+ZDP(JZ)*ZDP(JZ)
                YR4(JZ,IZ)=YS6(IZ)*YS6(IZ)+ZDM(JZ)*ZDM(JZ)


                AO(JZ,IZ)=EE*(YD6(IZ)/YR1(JZ,IZ)-
     1                    YD6(IZ)/YR2(JZ,IZ)-
     1                    YS6(IZ)/YR3(JZ,IZ)+
     1                    YS6(IZ)/YR4(JZ,IZ))
C                                                         SIX!

              END DO
              END DO
C             THIS COMPLETES FILLING AO(JZ,IZ),NOW GO ON TO AX(JZ,IZ,KX)

              DO KX=-NX+1,NX
                XX=X(KX)-X(-NX)
                XS=XX*XX

              DO IZ=1,NY2      !FLOOR
              DO JZ=1,NY2       !FLOOR
C                                                    ONE!
                R1=XS+YDP(JZ,IZ)*YDP(JZ,IZ)
                R1=SQRT(R1)
                R2=XS+YDM(JZ,IZ)*YDM(JZ,IZ)
                R2=SQRT(R2)
                R3=XS+YSP(JZ,IZ)*YSP(JZ,IZ)
                R3=SQRT(R3)
                R4=XS+YSM(JZ,IZ)*YSM(JZ,IZ)
                R4=SQRT(R4)
                AX(JZ,IZ,KX)=EE*XX*(1/R1*(YDP(JZ,IZ)*1/XS+1/YDP(JZ,IZ))-
```

```
     1          1/R2*(YDM(JZ,IZ)*1/XS+1/YDM(JZ,IZ))+1/R3*(YSP(JZ,IZ)*
     1          1/XS+1/YSP(JZ,IZ))-1/R4*(YSM(JZ,IZ)*1/XS+1/YSM(JZ,IZ)))
        END DO

        DO JZ=1,NZ2     !  WALL VORTEX

          R1=XS+ZY1(JZ,IZ)
          R2=XS+ZY2(JZ,IZ)
          R3=XS+ZY3(JZ,IZ)
          R4=XS+ZY4(JZ,IZ)

        AX(JZ,IZ,KX)=EE*XX*(1/R1*YD(IZ)/ZY1(JZ,IZ)-1/R2*YD(IZ)/
     1        ZY2(JZ,IZ)-1/R3*YP(IZ)/ZY3(JZ,IZ)+1/R4*YP(IZ)/
     1        ZY4(JZ,IZ))
C                                                        TWO

        END DO
        END DO

        DO IZ=NY2+1,NY2+NZ
        DO JZ=1,NY2

          R1=XS+YZ1(JZ,IZ)
          R1=SQRT(R1)
          R2=XS+YZ2(JZ,IZ)
          R2=SQRT(R2)
          R3=XS+YZ3(JZ,IZ)
          R3=SQRT(R3)
          R4=XS+YZ4(JZ,IZ)
          R4=SQRT(R4)
        AX(JZ,IZ,KX)=-EE*XX*ZD(IZ)*(1/R1/YZ1(JZ,IZ)-1/R2/
     1         YZ2(JZ,IZ)+1/R3/YZ3(JZ,IZ)-1/R4/YZ4(JZ,IZ))

C                                                            THREE!
        END DO

        DO JZ=NY2+1,NY2+NZ2

          R1=XS+ZDP2(JZ,IZ)*ZDP2(JZ,IZ)
          R1=SQRT(R1)
          R2=XS+ZDM2(JZ,IZ)*ZDM2(JZ,IZ)
          R2=SQRT(R2)
          R3=XS+ZDP3(JZ,IZ)
          R3=SQRT(R3)
          R4=XS+ZDM3(JZ,IZ)
          R4=SQRT(R4)

        AX(JZ,IZ,KX)=EE*XX*(1/R1*ZDP2(JZ,IZ)*(1/XS+1/ZDP2(JZ,IZ))-
     1       1/R2*ZDM2(JZ,IZ)*(1/XS+1/ZDM2(JZ,IZ))-1/R3*ZDP2(JZ,IZ)*
     1       (1/ZDP3(JZ,IZ)+1/(XS+D*D))+1/R4*ZDM2(JZ,IZ)*
     1       (1/ZDM3(JZ,IZ)+1/(XS+D*D)))
C                                                          FOUR
        END DO
        END DO

        XDD=XS+D*D
        DO IZ=NY2+NZ+1,NY+NZ
        DO JZ=1,NY2

          R1=XS+YD1(JZ,IZ)
```

```
           R1=SQRT(R1)
           R2=XS+YD2(JZ,IZ)
           R2=SQRT(R2)
           R3=XS+YD3(JZ,IZ)
           R3=SQRT(R3)
           R4=XS+YD4(JZ,IZ)
           R4=SQRT(R4)

          AX(JZ,IZ,KX)=-EE*XX*(YDP(JZ,IZ)/R1*(1/XDD+1/YD1(JZ,IZ))-
     1         YDM(JZ,IZ)/R2*(1/XDD+1/YD2(JZ,IZ))+YSP(JZ,IZ)/R3*
     1         (1/XDD+1/YSP(JZ,IZ))-YSM(JZ,IZ)/R4*(1/XDD+1/YSM(JZ,IZ)))
C                             '  FIVE'
          END DO

          DO JZ=NY2+1,NY2+NZ2

           R1=XS+YR1(JZ,IZ)
           R1=SQRT(R1)
           R2=XS+YR2(JZ,IZ)
           R2=SQRT(R2)
           R3=XS+YR3(JZ,IZ)
           R3=SQRT(R3)
           R4=XS+YR4(JZ,IZ)
           R4=SQRT(R4)

          AX(JZ,IZ,KX)=EE*XX*(YD6(IZ)*(1/R1/YR1(JZ,IZ)-1/R2/YR2(JZ,IZ))+
     1          YS6(IZ)*(1/R3/YR3(JZ,IZ)-1/R4/YR4(JZ,IZ)))

C                                        '  SIX
          END DO
          END DO
          END DO

C         ARRAY AX IS NOW FILLED! THE ARRAY A DISCUSSED IN THE REPORT
C         IS SYNTHESIZED FROM A0 AND AX TO SAVE SPACE WHEN A LARGE
C         NUMBER OF PANELS ARE USED.


          DO IX=-NX,NX
          DO IZ=1,NY+NZ
           LZ=NY+NZ+1-IZ
          DO JX=-NX,IX
           KX=-NX-JX+IX
          DO JZ=1,NY2+NZ2

           PHINO(IZ,IX)=PHINO(IZ,IX)+VX(JZ,JX)*(A0(JZ,IZ)+
     1          AX(JZ,IZ,KX))
          END DO

          DO JZ=NY2+NZ2+1,NY+NZ

           KZ=NY+NZ+1-JZ
          PHINO(IZ,IX)=PHINO(IZ,IX)+VX(JZ,JX)*(A0(KZ,LZ)+
     1         AX(KZ,LZ,KX))
          END DO
          END DO

          DO JX=IX+1,NX

           LX=-NX+JX-IX
```

```
      DO JZ=1,NY2+NZ2

      PHINO(IZ,IX)=PHINO(IZ,IX)+VX(JZ,JX)*(AO(JZ,IZ)-
     1   AX(JZ,IZ,LX))
      END DO

      DO JZ=NY2+NZ2+1,NY+NZ

        KZ=NY+NZ+1-JZ
      PHINO(IZ,IX)=PHINO(IZ,IX)+VX(JZ,JX)*(AO(KZ,LZ)-
     1    AX(KZ,LZ,LX))
      END DO
      END DO
      END DO
      END DO

C     THE PHINO ARRAY IS NOW FILLED!
C     REMEMBER THAT PHINO IS THE WALL-FREE NORMAL VELOCITY
C     CALCULATED FROM THE MEASURED  WALL PHI-X AND PHI-N
C     VALUES(MEASVX AND MEASVN).


      DO IX=-NX,NX
      DO IZ=1,NY+NZ

       VN(IZ,IX)=PHINO(IZ,IX)-MEASVN(IZ,IX)
C     THIS IS THE RESIDUAL EFFECT OF THE WALL PRESENCE. IF THESE
C     TWO VALUES WERE EQUAL WE WOULD HAVE PERFECT WALL ADAPTION
      END DO
      END DO

      OPEN(UNIT=3,NAME='PHINWALL.DAT',STATUS='NEW')
      WRITE(3,*) ((VN(I,J),I=1,NY+NZ),J=-NX,NX)
      CLOSE(3)


      END
```

C-2

Report Documentation Page

1. Report No. NASA CR-178374	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle An Engineering Study of Hybrid Adaptation of Wind Tunnel Walls for Three Dimensional Testing	5. Report Date December 1987 6. Performing Organization Code
7. Author(s) Clinton Brown, Kenneth Kalumuck, and David Waxman	8. Performing Organization Report No. 10. Work Unit No. 505-61-01-04
9. Performing Organization Name and Address Tracor Hydronautics, Inc. 7210 Pindell School Road. Laurel, MD 20707	11. Contract or Grant No. NAS1-18184 13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Perry A. Newman
Final Report

16. Abstract

The present report deals with solid wall tunnels having only upper and lower walls flexing. An algorithm for selecting the wall contours for both two and three dimensional wall flexure is presented and numerical experiments are used to validate its applicability to the general test case of three dimensional lifting aircraft models in rectangular cross-section wind tunnels. The method requires an initial approximate representation of the model flow field at a given lift with walls absent. The numerical methods utilized are derived by use of Green's source solutions obtained using the method of images; first order linearized flow theory is employed with Prandtl-Glauert compressibility transformations. Equations are derived for the flexed shape of a simple constant thickness plate wall under the influence of a finite number of jacks in an axial row along the plate centerline. The Green's source methods are developed to provide estimations of residual flow distortion (interferences) with measured wall pressures and wall flow inclinations as inputs.

17. Key Words (Suggested by Author(s)) Wall Interference Adaptive Walls Wall Interference Corrections	18. Distribution Statement Unclassified--Unlimited Subject Category 02

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 98	22. Price A05

NASA FORM 1626 OCT 86