PREDICTION OF FORCES AND MOMENTS FOR FLIGHT VEHICLE CONTROL EFFECTORS— FINAL REPORT

PART II: AN ANALYSIS OF DELTA WING AERODYNAMIC CONTROL EFFECTIVENESS IN GROUND EFFECT

NASA Grant NAG 1-849

The National Aeronautics and Space Administration
Langley Research Center
ATTN: Dr. John Shaughnessy
MS 494
Hampton, VA 23665

M. Maughmer, L. Ozoroski, T. Ozoroski, D. Straussfogel
Department of Aerospace Engineering
The Pennsylvania State University
233 Hammond Building
University Park, PA 16802

May 1990

Abstract

Many types of hypersonic aircraft configurations are currently being studied for their feasibility of future development. Since the control of the hypersonic configuratins throughout the speed range has a major impact on acceptable designs, it must be considered in the conceptual design stage. Part I of this report examines the ability of the aerodynamic analysis methods contained in an industry standard conceptual design system, APAS II, to estimate the forces and moments generated through control surface deflections from low subsonic to high hypersonic speeds. Predicted control forces and moments generated by various control effectors are compared with previously published wind tunnel and flight test data for three configurations: the North American X–15, the Space Shuttle Orbiter, and a hypersonic research airplane concept. Qualitative summaries of the results are given for each longitudinal force and moment and each control derivative in the various speed ranges. Results show that all predictions of longitudinal stabiltiy and control derivatives are acceptable for use at the conceptual design stage. Results for most lateral/directional control derivatives are acceptable for conceptual design purposes; however, predictions at supersonic Mach numbers for the change in yawing moment due to aileron deflection and the change in rolling moment due to rudder deflection are found to be unacceptable. Including shielding effects in the analysis is shown to have little effect on lift and pitching moment predictions while improving drag predictions. Overall, lateral/directional control derivatives show better agreement when shielding effects are not included.

In Part II of this report, an investigation of the aerodynamic control effectiveness of highly swept delta planforms operating in ground effect is presented. A vortex-lattice computer program incorporating a free wake is developed as a tool to calculate aerodynamic stability and control derivatives. Data generated using this program are compared to experimental data and to data from other vortex-lattice programs. Results show that an elevon deflection produces greater increments in C_L and C_M in ground effect than the same deflection produces out of ground effect and that the free wake is indeed necessary for good predictions near the ground.

Table of Contents

Page

Overview 1

Part I: Validation of Methods for Predicting Hypersonic Vehicle Controls Forces and Moments

Introduction 3

North American X-15 Research Aircraft 4

Low Speed: $M_\infty = .056$ 4

Transonic: $M_\infty = 0.80, 1.03, 1.18$ 4

Supersonic: $M_\infty = 2.96$ 7

Hypersonic: $M_\infty = 4.65, 6.83$ 8

Summary Results with Mach Number 9

Hypersonic Research Airplane 9

Low Speed: $M_\infty = 0.20$ 9

Transonic: $M_\infty = 0.80, 0.98, 1.20$ 10

Hypersonic: $M_\infty = 6.00$ 11

Summary Results with Mach Number 12

Rockwell Space Shuttle Orbiter 12

Low Speed: $M_\infty = 0.20$ 12

Transonic: $M_\infty = 0.80$ 14

Hypersonic: $M_\infty = 5.00, 20.0$ 14

Summary Results with Mach Number 16

Conclusions and Recommendations 17

References 19

Nomenclature 21

Tables 22

Figures 24

Appendix: Bibliography of Experimental Force and Moment Data For Hypersonic Vehicle Configurations

Table of Contents (cont'd)

Page

Part II: An Analysis of Delta Wing Aerodynamic
Control Effectiveness in Ground Effect

Introduction 1

Motivation for the Investigation 1

Scope of the Research 2

Review of Previous Work 3

Theoretical Predictions 3

Experimental Results 4

Theoretical Developments 6

The Vortex-Lattice Method 6

The Biot-Savart Law 11

Determination of the Influence Coefficients 13

Calculation of the Lift Coefficient 16

The Free Wake 18

The Ground Effect 23

The Suction Analogy 27

Total Lift 27

Total Pitching Moment 30

Flap Deflections 31

Results and Discussion 33

Verification of Results 33

VLM-FIG Predictions 49

Summary and Conclusions 64

References 66

Appendix A: Computer Program Description 69

Programming Method 69

Output 79

Appendix B: Computer Program Listing 80

Nomenclature 109

Part II: An Analysis of Delta Wing Aerodynamic Control Effectiveness in Ground Effect

Introduction

Motivation for the Investigation

Due to the new flowfield boundary conditions, like any aircraft, an aircraft with a highly swept delta planform will experience changes of its stability and control derivatives as it leaves or enters ground effect. Unlike other aircraft, however, takeoffs and landings are complicated since elevon deflections produce coupled changes of lift and pitching moments. It is therefore necessary to determine if there is sufficient control power to trim the pitching moment and allow flight at a desired lift coefficient when the stability and control derivatives are modified by ground effect. Such information is important during preliminary design so that control surface sizes can be estimated based on the least favorable flight conditions.

Because of the lift/moment coupling, elevon deflections required to trim can affect the lift in an adverse, though perhaps transient, manner. An example of this is the flare maneuver as the aircraft prepares to land. The elevon deflection necessary to rotate the nose upward causes a decreased camber of the wing which leads to a decreased lift coefficient. This results in an undesirable loss of altitude near the ground. As the nose rotates upward, lift is increased, altitude is regained, and the landing proceeds.

Given the proper stability and control derivatives as a function of height, it is possible to integrate the equations of motion to obtain the flight trajectory and determine if the desired trajectory is attainable.

Scope of the Research

A vortex-lattice-type computer program was written to facilitate the analysis of the effects of the ground. This program will be used as a tool to estimate the control and static derivatives in ground effect and out of ground effect. Once these are known, the equations of motion can be integrated and the flight trajectory found.

The vortex-lattice method is not capable of modelling thickness or viscous effects. Low aspect ratio delta wings must be thin for high-speed efficiency, however, so thickness effects may be neglected to the extent which linearized theory allows and the viscous effects of leading-edge separation may be accurately modeled by invoking the Polhamus Suction Analogy (Ref. 1–3). Since the program is intended for preliminary design purposes, it is required that it yields reliable results in a reasonable amount of time. A vortex-lattice program is therefore appropriate.

The program was named VLM-FIG, an acronym for Vortex-Lattice Method for analyzing Flaps In Ground effect. It is capable of analyzing wings which are twisted, cambered, or cranked at the leading and trailing edges. It can account for dihedral and can predict the effects of symmetrical leading- or trailing-edge flap deflections. If there are several flaps, each can be deflected independently.

The program incorporates a deformable wake and a provision to place a ground plane at a desired location below the wing. The time-dependent effects of the free wake and ground plane interactions are not considered. Although some recent research has shown the time rate of change of altitude to be critical for the accurate prediction of aerodynamic characteristics in ground effect, these claims are more true for aircraft which have extremely steep glideslopes than for supersonic transports or even glider-type hypersonic aerospace planes (Refs. 4,5).

The program will calculate the lift and moment coefficients in or out of ground effect, and with or without control deflections.

Review of Previous Work

Theoretical Predictions

Vortex-lattice computer programs have been used in the past to predict the potential flows over delta wings and have been combined with the suction analogy to predict the aerodynamic characteristics of delta wings (Ref. 6–9). These programs have been shown to be good predictors of delta wing characteristics, and have also been shown to be capable of predicting the low-speed aerodynamic characteristics of hypersonic wing-body combinations (Ref. 10). If the planform is cropped and/or cranked, then certain correction factors can be applied to the suction analogy to improve its predictive capabilities (Ref. 11–13).

Carlson et al. (Ref. 14–18) have developed vortex-lattice programs to predict the flap effectiveness for wings with near-delta planforms. The programs deal with only attached flow and they predict that the flap efficiency is highest when both the leading- and trailing-edge flaps are deflected. The programs use a flat wake approach and do not consider the effects of the ground. VLM-FIG has similar capabilities but is also able to predict the control power near the ground and considers the influence of a relaxed wake.

Fox (Ref. 19) uses a vortex-lattice code with a flat wake to obtain ground effect data for delta wings. The program predicts the experimental lift coefficient data well, but pitching moment and control power are not evaluated. More recently, Nuhait and Mook (Ref. 20) have investigated delta and non-delta planforms using an unsteady ground effect vortex-lattice program. It was found that for the steady case, aerodynamic coefficients generally increase as the wing nears the ground. The time rate of change of altitude enhances this effect, and greater aerodynamic increases were caused by greater sink rates. Thus, aircraft with steep glideslopes are more affected by time-dependent effects.

Experimental Results

During the early 1960's work was continuing on the development of the Concorde and, consequently, a systematic study of the effect of the ground on the aerodynamics of delta wings was conducted by Peckham (Ref. 21). Complete pitching moment and lift data was obtained for a series of delta wings with sharp edges, but no control deflections were included.

Chang and Muirhead (Ref. 4) conducted an experimental investigation of the time-dependent ground effects on delta wings. At high angles of attack and low ground heights, the unsteady effects nearly double the effect of the ground. The data was generated at a sink rate of $\dot{h}/V = 2.0$, however, which corresponds to a highly unrealistic glideslope angle of more than 60°.

There is a void in the literature for the subject of delta wings with control deflections in ground effect. Apparently, no studies similar to Peckham's have been conducted using delta wings with flaps in ground effect. Coe and Thomas (Ref. 22), however, have used an arrow-wing configuration for such a study, and the general trends of increasing aerodynamic coefficients with decreasing ground height were again observed.

Theoretical Developments

The Vortex-Lattice Method

From the Kutta-Joukowski Theorem, it is known that the lift of an airfoil is uniquely determined by the amount of circulation which it generates.

$$l = \rho V \gamma \tag{2.1}$$

where γ is the vortex strength per unit span, the span being in the direction of the axis of the vortex.

It can be easily shown using potential flow theory that an infinite vortex sheet with a constant strength per unit length in the plane of the vortex γ_D will induce a velocity of

$$v = \frac{\gamma_D}{2} \tag{2.2}$$

If such a distributed vortex is placed in an appropriately oriented free stream, the velocity above the vortex will add to the free stream velocity, while that below will subtract.

Because Bernoulli's Principle states that the difference in the velocities of this flowfield will create a pressure difference, this distributed vortex will experience a self-induced force which acts normal to the free stream. This force will have a magnitude which can be found by beginning with the Bernoulli equation,

$$P_1 + \frac{1}{2}\rho V_1^2 = P_2 + \frac{1}{2}\rho V_2^2$$

$$\Delta P = \frac{1}{2}\,\rho\,(V_2^2 - V_1^2)$$

where the subscripts 1 and 2 represent the location below and above the vortex, respectively. From Equation 2.2

$$V_1 = V - \frac{\gamma_D}{2}$$

$$V_2 = V + \frac{\gamma_D}{2}$$

Substitution leads to

$$\Delta P = \frac{1}{2}\,\rho\,[(V - \frac{\gamma_D}{2})^2 - (V - \frac{\gamma_D}{2})^2]$$

which reduces to

$$\Delta P = \frac{1}{2}\,\rho\,(2V\gamma_D)$$

To convert the pressure to a force, it must be multiplied by an area of $1 \cdot dx$. This results in

$$dl = \Delta P dx = \frac{1}{2}\,\rho\,(2V\gamma_D)dx$$

Using the relation

$$\gamma = \int_0^1 \gamma_D dx \tag{2.3}$$

and integrating both sides over a unit chord length, the above yields Equation 2.1, the Kutta-Joukowski equation (Ref. 23). Thus, a distributed vortex in a free stream can be used to model the flow over a thin airfoil.

Since the preceding approach to estimating the lift includes only vortices, there is no provision to account for the effects of thickness on the airfoil aerodynamic characteristics. For thin airfoils, which are necessary to reduce the wave drag of supersonic or hypersonic aircraft, this should not impede an accurate solution. In addition, this approach does not restrict the airfoil to lie on a straight line, since each segment may be linked to another at a different angle. If more that one segment is present, however, the lift cannot be found so simply. Because the vortex is no longer an infinite line, the induced velocities above and below the vortex become unknown.

Supposing that this problem can be overcome, then the force normal to a plane containing the vortex and inclined to the free stream at an angle α, can be determined by the equation

$$l = \rho\,(V\cos\alpha)\gamma_D \tag{2.4}$$

A force will also be generated parallel to the plane due to the addition and subtraction of vortex-induced velocities to the free-stream velocity component $V\sin\alpha$, which is given by

$$l = \rho\,(V\sin\alpha)\gamma_D \tag{2.5}$$

This force may be thought of as the leading-edge suction force, which is predicted by thin airfoil theory as a singular velocity at the leading edge of the airfoil. This singularity causes an infinitely low pressure acting over an infinitely thin leading-edge radius to produce a force which exactly cancels the drag due to the lift which acts normal to the tilted airfoil. In this way, D'Alembert's paradox remains satisfied.

Since an airfoil does not produce lift by generating a constant rectangular velocity field over its surfaces; a better approximation for the flow over an airfoil can be obtained using a series of linked vortices of differing strengths. Also, since a vortex sheet can be related to a point vortex by Equation 2.3, the airfoil may be further discretized by replacing each vortex sheet with a point vortex of equal total strength. Each of these point vortices is placed at the one-quarter chord point of its respective elemental chord, in accordance with Prandtl's standard lifting line theory.

In a similar fashion, two or more chordwise strips of vortices placed side-by-side and parallel to one another may be used to model a wing. Figure 2.1 shows how such a lattice of vortices can represent a wing. One important distinction between the two-dimensional airfoil and the three-dimensional wing is that the vortices of the wing must be horseshoe shaped to satisfy the conservation of vorticity. The continuous variation of vorticity in both the chordwise and the spanwise directions on the finite, three-dimensional wing is approximated by this lattice of horseshoe vortices.

Once the wing has been divided into the desired number of trapezoids, referred to as panels, with a horeshoe vortex bound to the quarter-chord line of each panel and filaments extending back along the sides of the panel to infinity, the procedure for determining the aerodynamic characteristics of the wing can be developed.

The distribution of the force acting on the wing surface is necessary since

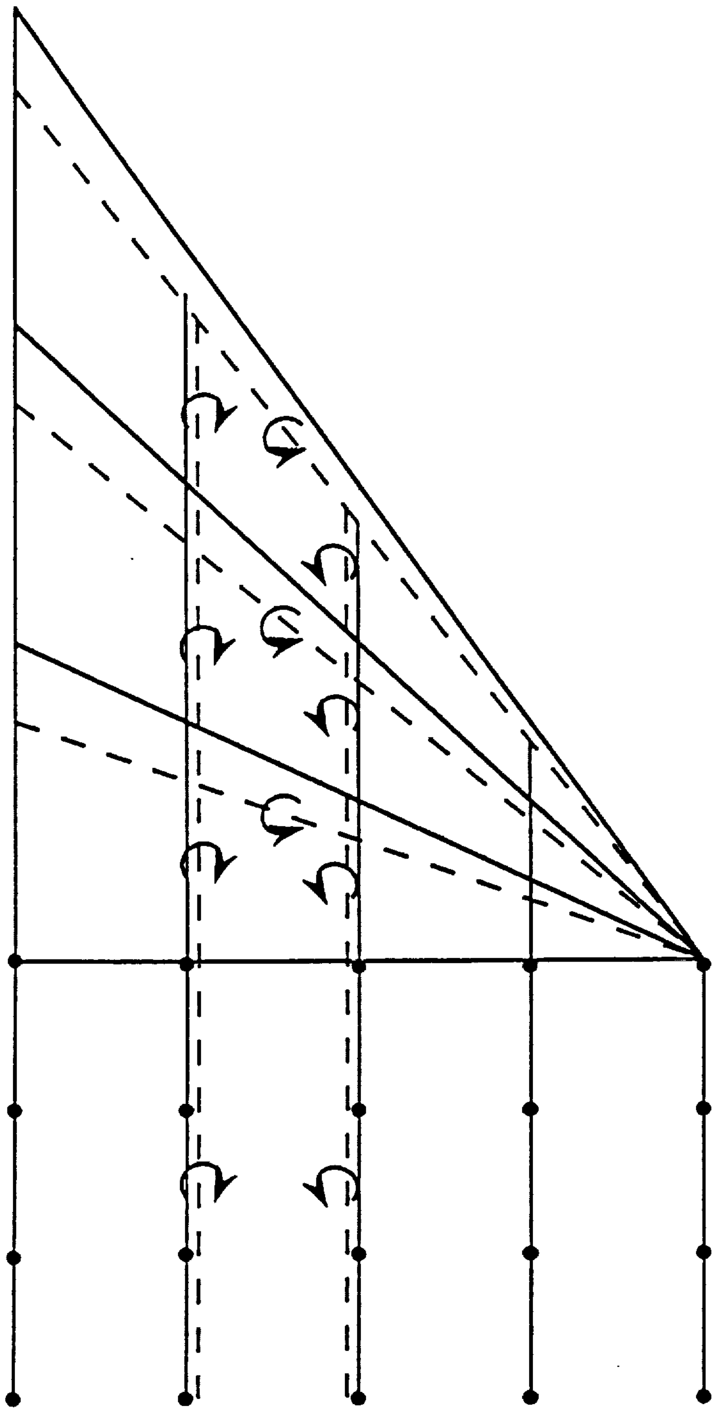

Figure 2.1 Vortex Lattice Arrangement Representing a Delta Wing and Its Wake. (Only One Row of Horseshoe Vortices Is Shown for Clarity)

it determines the aerodynamic characteristics. To determine this distribution, the velocities induced by each vortex must be found. A simplified approach is no longer applicable, since the flow is three dimensional and since each vortex has a different strength. To determine the velocities, the effect which each vortex has on the velocities near itself as well as on every other vortex must be considered.

The Biot-Savart Law

The Biot-Savart law can be used to determine how the point vortex will affect the velocity over its own panel and how it will effect velocities over the other panels, too. The equation can be written in its most general form as

$$\overrightarrow{dV} = \frac{\Gamma(\overrightarrow{dl} \times \overrightarrow{r})}{4\pi |\overrightarrow{r}|^3} \tag{2.6}$$

Since each horseshoe vortex consists of a left segment, a bound segment, and a right segment, this equation must be applied to each segment and then summed to obtain the total effect of one complete horseshoe vortex. If the endpoints of a certain segment are A and B, and the induced velocity is desired at point C, as shown in Figure 2.2, then the expression for the velocity can be written in the form:

$$\overrightarrow{V}_C = \frac{\gamma_n}{4\pi r_p}(\cos\theta_1 - \cos\theta_2) \tag{2.7}$$

where

$$r_p = \frac{\overrightarrow{r}_1 \times \overrightarrow{r}_2}{r_o} \tag{2.8a}$$

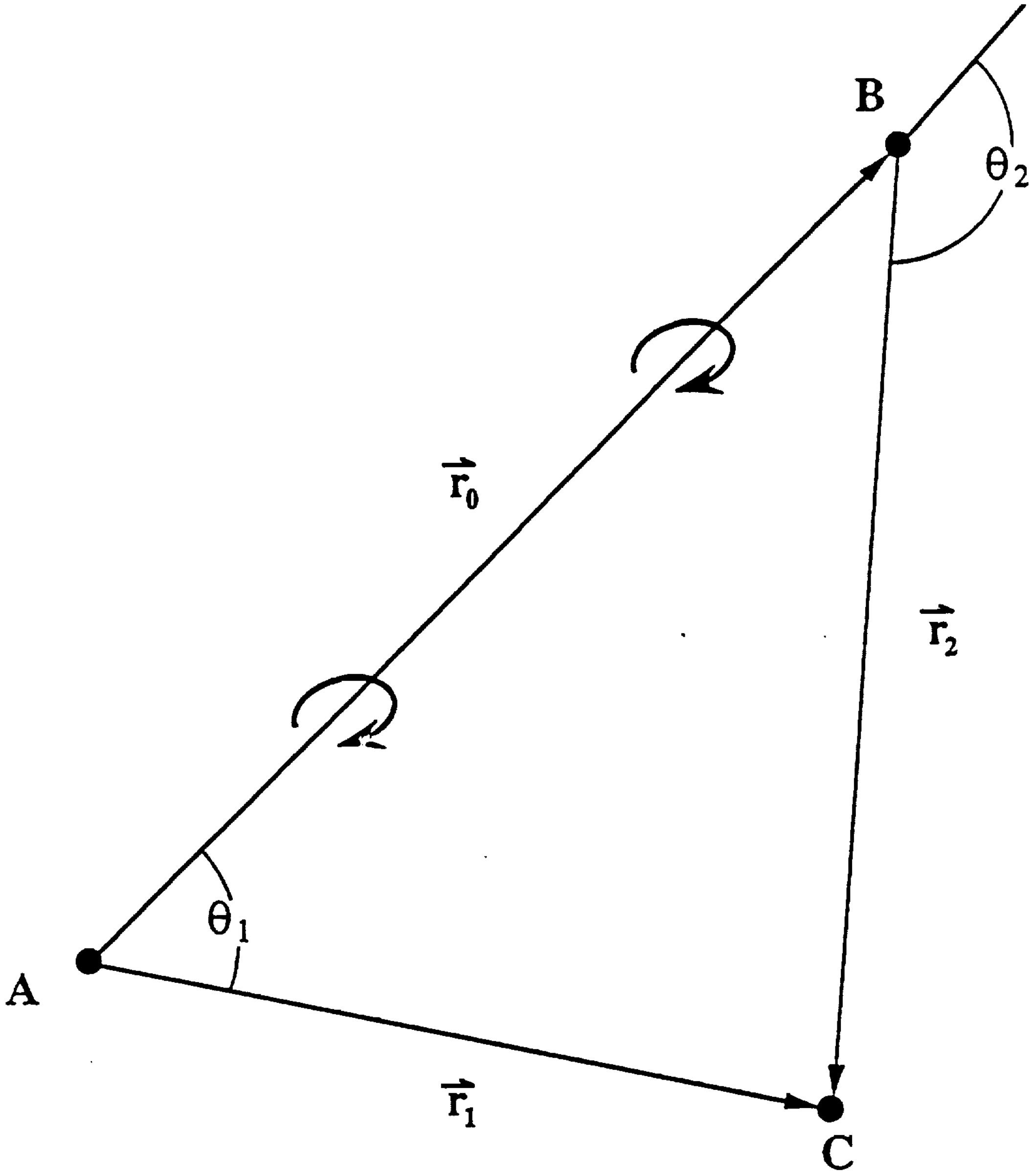

Figure 2.2 The Geometrical Arrangement for Applying the Biot-Savart Law to a Vortex Segment with Positive Vorticity Directed from A to B

$$\cos\theta_1 = \frac{\vec{r}_o \cdot \vec{r}_1}{r_o r_1} \tag{2.8b}$$

$$\cos\theta_2 = \frac{\vec{r}_o \cdot \vec{r}_2}{r_o r_2} \tag{2.8c}$$

Equation 2.7 is a vector equation. If the points are arranged according to Figure 2.2 and the vortex has a positive sign and points in the direction from A to B, then the algebraic signs of the induced velocity will be accounted for properly (Ref. 24). Care must be taken to insure that the vortex points in the proper direction or the sign of the induced velocity will be opposite of what it should be.

Determination of the Influence Coefficients

The velocity which a particular horseshoe vortex induces at every other panel can be found by summing the velocity induced by its left, bound, and right segments at the point C, which varies to represent the location of different panels. The location of point C on a given panel is the three-quarter chord point of the panel and it is centrally located in the spanwise direction of the panel. This point is referred to as the control point of the panel. Thus, every horseshoe vortex influences every control point on the wing according to Equation 2.7. Since the individual γ_n are not yet known, only the influence coefficients, represented by the factor

$$C_{mn} = \frac{1}{4\pi r_p}(\cos\theta_1 - \cos\theta_2) \tag{2.9}$$

can be calculated.

In addition to the coefficients from one side of the wing, the coefficients from the other side of the wing must be included. This does not double the number of unknowns since the flowfield is symmetric; however, the same procedure to calculate influence coefficients at every control point due to every vortex must be executed. This doubles the number of calculations and, since this represents the largest fraction of the calculations, the CPU time will nearly double.

If the flowfield were not symmetric, the number of unknowns would double and the flowfield would need to be solved as a single entity. This would require storing all of the locations of both the left and right side vortices. In the case of a symmetric flowfield, the vortices on opposite sides of the wing centerline can be treated as mirror images, but located at different distances from all the control points on the right side. This is illustrated in Figure 2.3. This has the simplifying effect of allowing the coordinates of the right half to be used to calculate the influence coefficients of the left half merely by changing the sign of the y-coordinates of the segment endpoints. Note that points A and B are interchanged since the sign of the vorticity is opposite.

Now, the influence coefficients from each vortex γ_n are known at each control point, C_m. This represents a set of simultaneous equations which can be written

$$\overrightarrow{V}_{mn} = \overrightarrow{C}_{mn}\gamma_n \tag{2.10}$$

where n represents a particular vortex and m represents a particular control point.

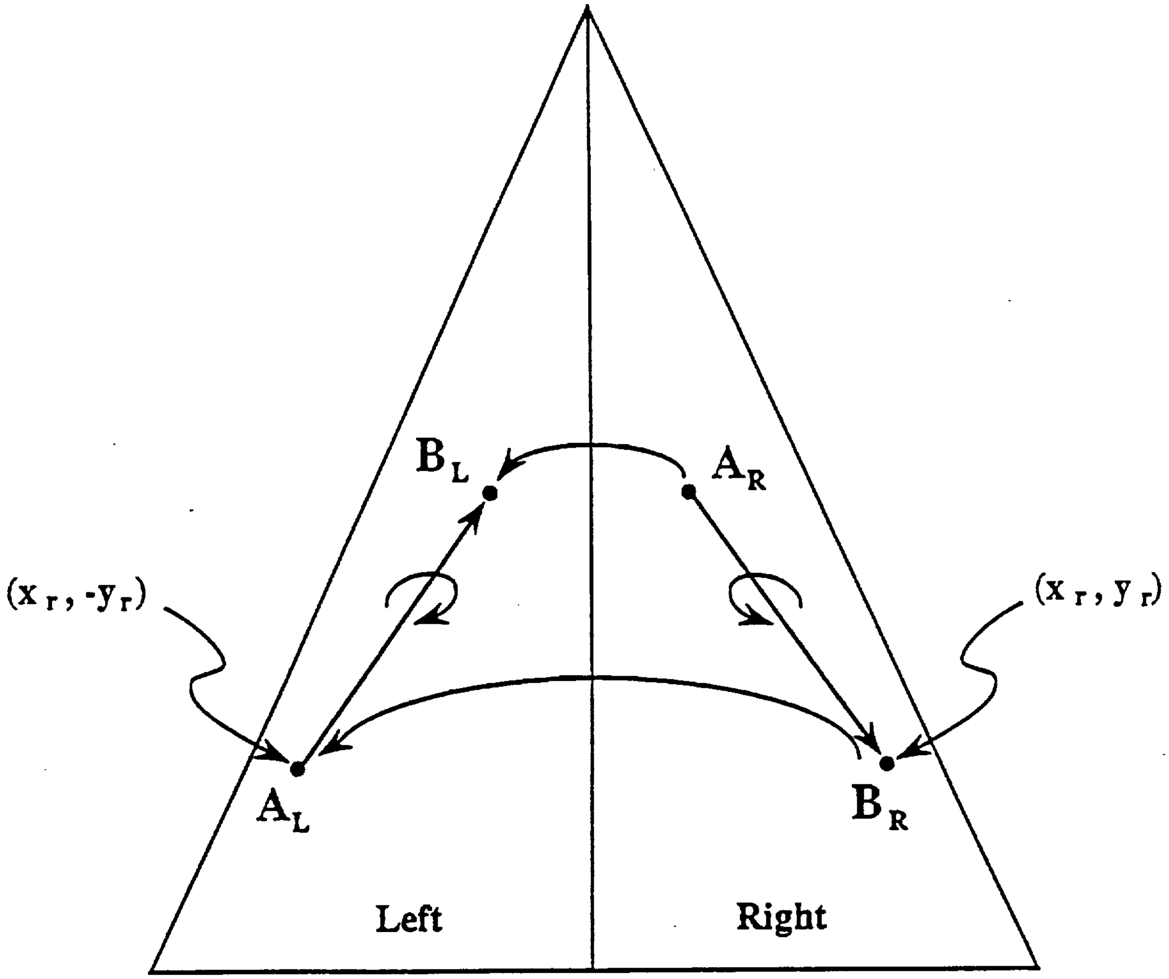

Figure 2.3 Reflection Over the Mid-Chord Line to Define Points on the Left Portion of the Wing

All that must be determined are the strengths of the individual vortices. These are found by requiring that the free stream flow perpendicular to the wing at each control point does not pass through the wing but, rather, is blocked by being equal but in the opposite direction to the induced velocity perpendicular to the wing at each control point. The solution is a function of the angle of attack, the dihedral angle, and the local slope in the chordwise direction. For a flat wing with no dihedral, it is only a function of the angle of attack at each control point as given by

$$w = V \sin \alpha \tag{2.11}$$

Departures from this simplified case are small for thin supersonic wings which have little camber (Ref. 6–9).

With the velocities, boundary conditions, and influence coefficients known, the strengths of all the vortices can be found. This is the most important step of the vortex-lattice method. Once the strengths of the vortices are known, the flowfield is solved and the aerodynamic properties of the wing can be found.

Calculation of the Lift Coefficient

The lift coefficient of an airfoil can be calculated as follows. Beginning with the definition of sectional lift coefficient,

$$l = \frac{1}{2} \rho V^2 c C_l \tag{2.12}$$

and equating the left side to Equation 2.1 yields

$$\frac{1}{2}\rho V^2 c C_l = \rho V \gamma \tag{2.13}$$

Without loss of generality, the free stream velocity may be set equal to 1. Cancelling the density results in a equation which relates the lift coefficient directly to the circulation

$$C_l = \frac{2\gamma}{c} \tag{2.14}$$

For a three-dimensional wing, the analysis needs to be generalized a bit more. The spanwise sectional lift coefficients are summed using a weighted average of the local chord times the local lift coefficient. Consider a wing which has continuously varying lift and a continuously varying chord in the spanwise direction. The wing will be discretized depending on the spanwise number of horseshoe vortices which is chosen to represent the wing. Such a wing is shown in Figure 2.1. The wing lift coefficient can be obtained from Equation 2.14 by summing the sectional lift coefficients across the span and weighting each by the amount of wing area which its panel covers. Allowing the subscript j to represent a particular spanwise location, and summing across the span leads to

$$C_L = \frac{1}{S}\sum_{j=1}^{\mathrm{NSPAN}} c_j C_{l_j} \Delta y$$

Substituting Equation 2.14 into this expression yields

$$C_L = \frac{1}{S}\sum_{j=1}^{\mathrm{NSPAN}} 2\gamma_j \Delta y \tag{2.15}$$

where

$$\gamma_j = \sum_{i=1}^{\text{NCHORD}} \gamma_i \qquad\qquad j = \text{constant}$$

Multiplying by two to account for the lift acting on both sides of the wing yields

$$C_L = \frac{4\Delta y}{S} \sum_{j=1}^{\text{NSPAN}} \gamma_j(y) \tag{2.16}$$

It is interesting to note that the effect of weighting the lift coefficient by the panel area is to eliminate the local chord term from the final equation. The only term with chordwise or spanwise dependence remaining in Equation 2.16 is the circulation term. This allows for a slightly simpler summation procedure. The method for calculaton of the pitching moment follows from this derivation and will be discussed toward the end of this chapter.

The Free Wake

The standard vortex lattice consists of horseshoe vortices which each trail two filaments, at some fixed angle, from the bound vortex to infinity. This is referred to as a flat or fixed wake method. In this method, the wake will support a pressure difference across its boundary and alter the free-air aerodynamic coefficients. At high angles of attack and for low aspect ratios, the wake becomes more deformed, implying that a flat wake approach will support greater pressure differences and cause anomolies in the aerodynamic predictions.

The standard approach also causes difficulty when a ground plane is introduced for obtaining the ground effect characteristics. Since the filaments trail downward towards the ground plane, they will intersect and pass below the ground– a serious misrepresentation of reality. Furthermore, the image filaments below the ground will pass up and through the ground into the real world resulting in another serious misrepresentation. Since these problems are pertinent particularly to this investigation, a free wake analysis was used. The final geometry of the free wake was determined using the following procedure.

The wake location is determined by using an iterative process which calculates the wing lift distribution, moves the wake, calculates a new lift distribution, and moves the wake again. This continues until the entire wake converges to within a specified criterion.

The geometry of the wake and trailing filaments are shown in Figure 2.4. The trailing filaments are subdivided into several segments on both the wake and the wing. This allows the vortex filaments to more closely approximate a curved path in the wake, or the camber of the wing. This does not have any effect on the strength of the vortices, it simply allows flexability of location.

Each segment endpoint, except for the last one, is a node at which downwash and sidewash velocities are calculated. The velocities are found by using the Biot-Savart law, just as are the influence coefficients at the control points on the wing. For the wake, however, the strength of each vortex is known, so the velocities may be directly calculated without solving a set of simultaneous equations. The velocity at a node determines the new wake location at that node. Before the

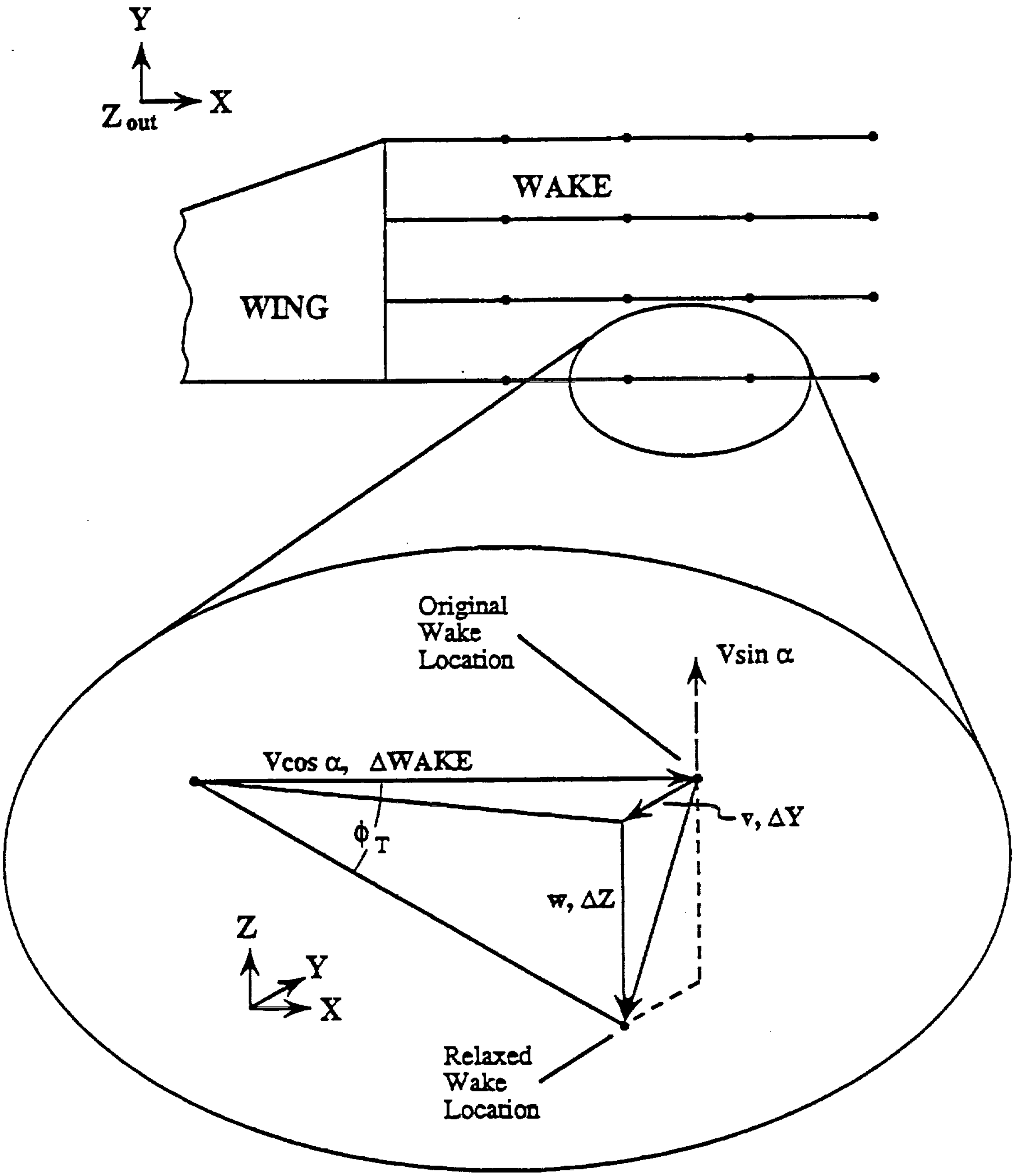

Figure 2.4 Vorticity Segments in the Wake and the Geometry Used for Moving Them After an Iteration (u and ΔX Are Not Shown).

velocities are calculated for the first time, each node in the wake shares the same y- and z-coordinates as the point on the wing trailing edge over which it passes. Each successive node behind the trailing edge will have an x-coordinate which is Δwake greater. This has the effect of preserving the sweep angle and shape of the trailing edge throughout the wake.

From the magnified portion of the wake in Figure 2.4, it is easy to see that

$$\Delta x = \frac{u}{V}\,\Delta\text{wake} \approx 0 \tag{2.17a}$$

$$\Delta y = \frac{v}{V}\,\Delta\text{wake} \tag{2.17b}$$

$$\Delta z = \frac{w + V\sin\alpha}{V}\,\Delta\text{wake} \tag{2.17c}$$

In words, the free stream velocity acting over a certain time divided by the induced x, y, and z velocities acting over the same time will yield a vector which points directly towards the new wake location. It is necessary to add the z component of the free stream to the velocities in the wake since no solid surface prevents flow through the plane of the original wake. When this is multiplied by Δwake, the new wake location is determined and the node can be moved. Thus, the induced velocities at the node just ahead of node i must be used to determine the new location of node i. So although the trailing edge is fixed, the velocities must be calculated there for the next wake node to be properly moved. Similarly, this explains why it is unnecessary to calculate the induced velocities at the final wake node.

The wake nodes of each iteration can be compared to the locations found by the previous iteration or, equivalently, they can be compared to the locations of fixed points in the wake. This investigation uses the second approach since the programming is easier to implement, and uses the original wake locations as the fixed locations in the wake. Reference to Figure 2.4 allows a comparison of the type

$$\phi_{T_{\substack{\text{current}\\\text{iteration}}}} - \phi_{T_{\substack{\text{previous}\\\text{iteration}}}} < \begin{matrix}\text{specified}\\\text{convergence}\\\text{criterion}\end{matrix} \tag{2.18}$$

There are several methods by which the wake iteration may be terminated. The distance that the wake will move is not known *a priori*, since the value of Δwake is arbitrary. For this reason, it was decided that the wake convergence should be measured in terms of angular distance. Equivalent downwash velocities at different locations in the wake will result in different angles, if they are measured with respect to a line emanating from a single location such as the wing trailing edge. It is conceivable that for points far back in the wake such an approach could lead to a "trigonometric convergence" where even large wake movements would be dwarfed by the node's distance to the trailing edge. If angles are measured with respect to appropriate and different points for each node, however, then angular distances are as reliable as linear distances and the *a priori* knowledge of the magnitude of Δwake is not required. These appropriate points are the original wake nodes and the angle which is measured is denoted in Figure 2.4 as ϕ_T. Two successive iterations generate two values of ϕ_T per node and if all of these values are within say, 1/100 of a degree, the iteration is terminated.

The Ground Effect

The effect of ground proximity is modelled by placing an identical image wing the same distance below the ground as the real wing is above the ground. The sign of the circulation of the image wing is equal in magnitude but opposite in direction to that of the real wing. This has the effect of switching points A and B in Figure 2.2. The geometry of the two wings near the ground is shown in Figure 2.5.

Finding the coordinates of the vortex segments which lie on the image wing and in the image wake is accomplished similarly to the previously discussed addition of the left half wing. Because the wing is at an angle of attack, however, certain transformations must be used to determine the coordinates of the image wing. For very small angles of attack, these transformations vanish since the intersection of the extensions of the two wings is approaching a point infinitely far downstream. If this were the case, the coordinates of the image wing would be determined by switching signs of the y- and/or z-coordinates of the real wing, depending on the particular location of the image segment. As shown in Figure 2.6 and using Region 1 for comparison, Region 2 would have opposite y-coordinates, Region 3 would have opposite y- and z-coordinates and Region 4 would have opposite z-coordinates.

Since the image wing coordinates are found by rotating the real wing through

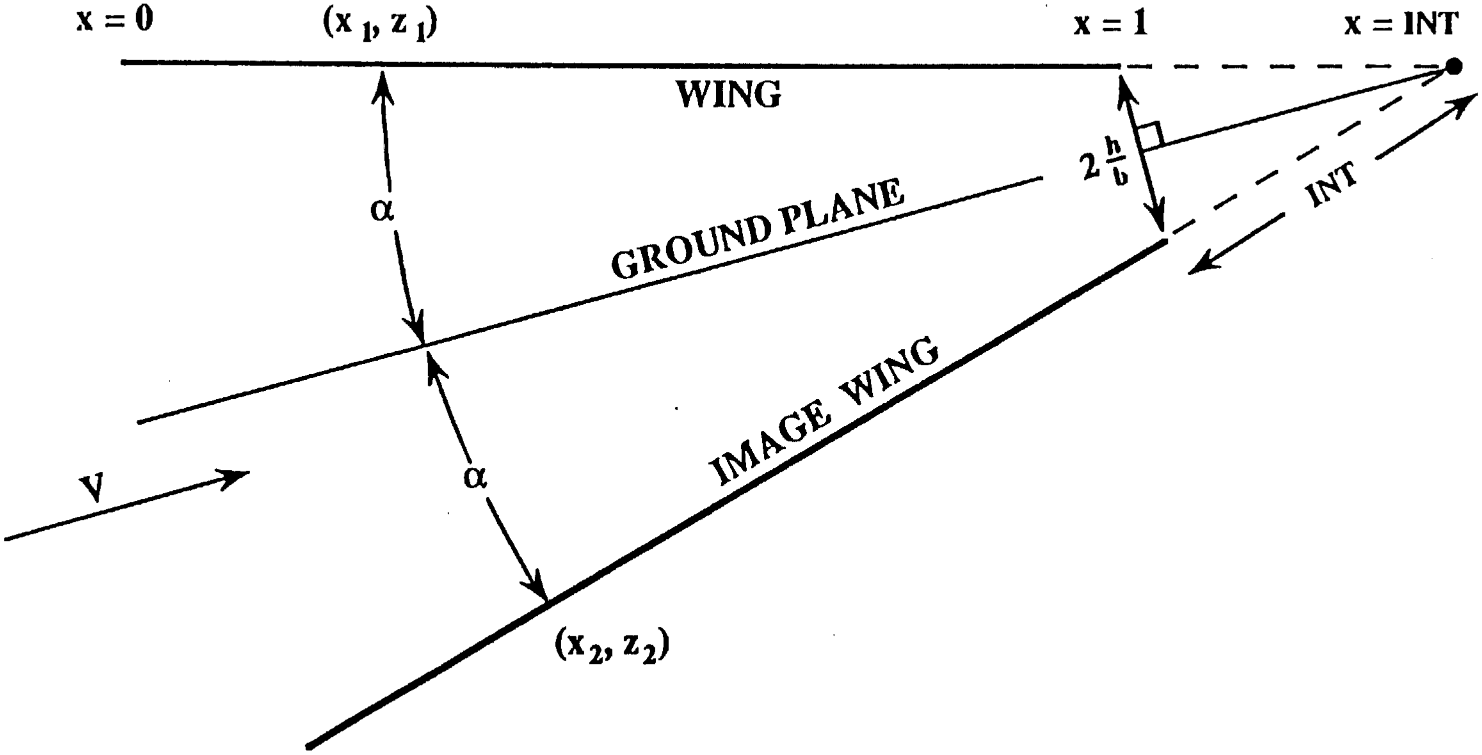

Figure 2.5 Geometry of a Wing and an Image Wing When in Ground Proximity

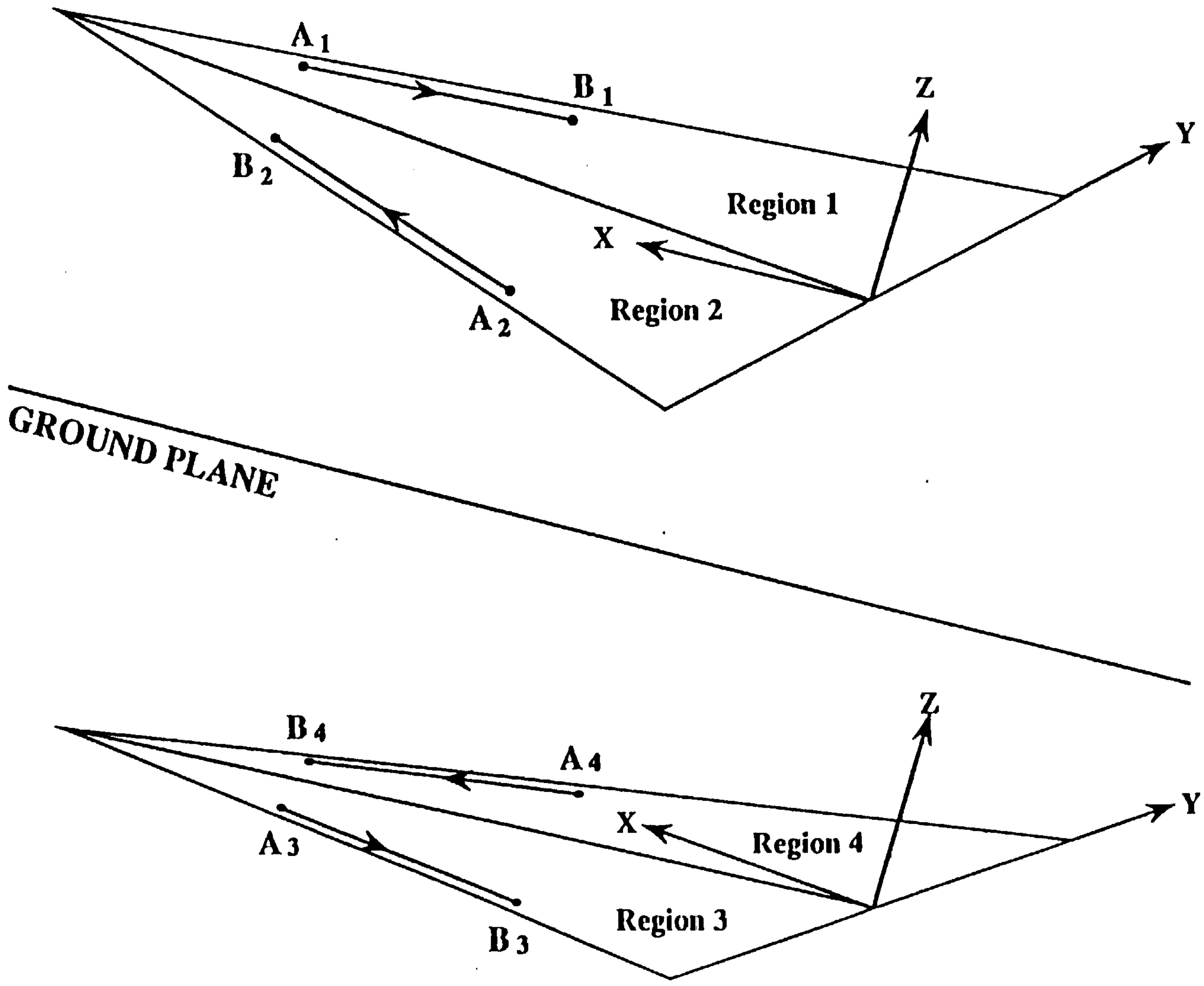

Figure 2.6 Relation of Various Points in a Region Relative to the Points in Region 1.

an angle of 2α, such a simplified approach can not be used if α is not approximately zero. Instead, the z-coordinates of Region 3 and Region 4 must undergo transformations rather than sign changes. This z-coordinate transformation depends on h/b and the additional height above the ground caused by camber, dihedral, and twist. The changes in the signs of the y-coordinates are, however, the same as those changes which would be made when considering the illustrative simplified case.

The transformations of the coordinates for the general case of an angle of attack which is not approaching zero are as follows. First, the point of intersection of the downstream extension of the mid-chords of the two wings must be determined. If the height of the wing above the ground is measured from the trailing edge and is non-dimensionalized by the wing span, the intersection point, INT, is found by

$$\mathrm{INT} = \frac{h/b}{\sin\alpha} \tag{2.19}$$

The z-coordinates of the image wing are also found using simple geometry by the equation

$$z_2 = z_1 - (\mathrm{DIST})\sin(2\alpha) \tag{2.20}$$

where

$$\mathrm{DIST} = \left[(\mathrm{XTE}(I) - x_1) + \mathrm{INT}\right]\cos(2\alpha) \tag{2.21}$$

The new x-coordinates are then found by the equations

$$x_2 = \mathrm{INT}' - \mathrm{DIST} \tag{2.22}$$

where

$$\mathrm{INT}' = \mathrm{XTE}(I) + \mathrm{INT} \tag{2.23}$$

If the non-dimensional root chord has been set to 1, then $\mathrm{XTE}(I) = 1$ for Equations (2.21) and (2.23). If the trailing edge is notched to create an arrow-type planform, it is important to determine the value of the root chord h/b which corresponds to the wing tip intersection with the ground. This is the minimum h/b achievable for that planform.

In order to compare the h/b values obtained from sources which use points other than the trailing edge to reference the wing height above the ground, the following conversion formula is used

$$h = h_{\mathsf{source}} - \left(c_o - c_{\mathsf{REF}}\right) \sin \alpha \tag{2.24}$$

c_o can be set to 1 as it has been for other calculations.

The Suction Analogy

Total Lift

For highly swept, low aspect ratio wings, potential flow theory is not sufficient for calculating the aerodynamic characteristics. Above a small angle of attack, which is approximately 4 degrees for thin supersonic or hypersonic wings, the potential flow separates from the leading edge and creates a strong vortex above the wing along the leading edge. The vortex will contribute to the lift and the drag of the wing and as the angle of attack increases, the vortex becomes stronger. At angles of attack greater than about 20 degrees, the part of the vortex near the trailing edge will begin to migrate inboard as well as upward and away from the back of wing.

In order to calculate the affects of this vortex, the Polhamus Suction Analogy is used. The method is based on the premise that when the flow separates from the leading edge, the leading-edge suction force is rotated 90° to become a lift force rather than a thrust force. It is also assumed that the flow reattaches to the wing downstream of the leading-edge vortex and implicitly requires a Kutta-type boundary condition at the leading edge. An interesting feature of this method is that it uses potential flow theory to calculate the non-linear vortex effects.

According to the suction analogy, the lift coefficient of a delta wing is given by

$$C_L = K_p \cos^2\alpha \, \sin\alpha + K_v \cos\alpha \, \sin^2\alpha \tag{2.25}$$

where the first term represents the potential lift and the second term represents the leading-edge vortex lift. The first term does not include the component of the leading-edge suction force which acts in the positive z-direction, rather, this term is included in the vortex lift term.

The factor K_p is determined by calculating the spanwise sectional lift coefficients, dividing them by $\cos^2\alpha \, \sin\alpha$, and then summing them across the span. For small angles of attack, $\cos\alpha \approx 1$, $\sin\alpha \approx \alpha$, and $\sin^2\alpha \approx 0$. In this case,

Equation (2.25) reduces to

$$C_L = K_p \alpha$$

which shows that K_p is simply the potential lift curve slope. For a delta wing of given aspect ratio, K_p is constant to first order. When the effects of a deformed wake are included, it becomes a weak function of the angle of attack. It is also a function of the wing height above the ground.

The factor K_v is determined by applying the Kutta-Joukowski theorem to the flow around the leading edge of the airfoil. This approach yields

$$C_t = \frac{2(\sin\alpha - \sin\alpha_i)\gamma}{c} \tag{2.26}$$

which is the same form as Equation 2.14. Following a similar procedure as was used to calculate C_L for the entire wing, C_T for the entire wing is given by

$$C_T = \frac{4\Delta y}{S} \sum_{j=1}^{\text{NSPAN}} \gamma_j \left(\sin\alpha - \sin\alpha_{\text{IN}}\right)_j \tag{2.27}$$

The suction force depends on the local leading-edge sweep angle and is given by

$$C_S = \frac{C_T}{\sin\Lambda} \tag{2.28}$$

The vortex lift, C_{Lv}, and the vortex constant, K_v, are then given by

$$C_{Lv} = C_S \cos\alpha \tag{2.29}$$

$$K_v = C_{Lv}/\sin^2\alpha\,\cos\alpha \tag{2.30}$$

Like K_p, K_v is constant to first order, but is a function of the angle of attack when the effects of the deformed wake are included. It is also a function of the wing height above the ground.

Total Pitching Moment

The potential pitching moment is calculated by finding the normal force which acts on a particular panel and multiplying it by the panel distance to the wing apex. This is repeated for each panel and then summed to obtain the total moment. Note that because of the loss of leading-edge suction, C_N is given by the equation

$$C_N = C_L / \cos\alpha \tag{2.31}$$

The vortex lift causes a moment also, but the method for finding it is somewhat different. Since the flow separates at the leading edge, the vortex lift acts close to the leading edge across the entire span. Thus, vortex lift is almost a force per unit length of leading edge rather than a force which is distributed over all of the panels of the wing, as is the potential lift. The leading-edge suction force is calculated at a particular leading-edge segment to the apex. It is assumed that only vortices in the chordwise row of the leading-edge segment influence the suction force at that segment.

In order to compare data from references which use different pitching moment axes and different reference chords, the following conversion is used

$$C_M = \left[\frac{\bar{c}}{c_o} (C_M - C_{M_o}) \right]_{\text{source}} - \frac{C_L}{\bar{c}/c_o} (\bar{c}_{\text{moment}}) \tag{2.32}$$

Normally, the sign of C_M from the source will be negative.

Strictly, the suction analogy is applicable only to delta planforms. For arrow-type planforms, there may not be enough outboard wing area for the flow to reattach as the analogy requires. For diamond-type planforms, there is area behind the basic delta shape which allows for additional lift, which is mainly potential. The effect of the notch ratio on the lift coefficient is small; however, the effect of the notch ratio on the pitching moment coefficient is large since the notch is located at the rear of the wing. This provides a large moment arm for small changes in lift and results in poor predictions for non-delta planforms.

The wing leading edge may be cranked and the wing tip may be cropped. If this is the case, there are additional vortex constants which account for the lift generated around the side edge and for the downstream persistence of a shed vortex due to a change in the leading-edge vortex strength caused by the crank.

Flap Deflections

When a flap is deflected, it changes the locations of the vortex segment endpoints and the control points which lie on panels which are affected by the deflection. Also, the slope of the wing at the affected control points is changed by an amount equal to the flap deflection angle. These changes in location and slope alter the influence coefficients and boundary conditions, resulting in a different flowfield.

Results and Discussion

Verification of Results

Figures 3.1–3.10 are presented to provide a validation of the ability of VLM-FIG to predict experimental results. They are also intended for comparison of VLM-FIG to other programs, primarily the Aerodynamic Preliminary Analysis System, APAS (Ref. 25) which is a fixed-wake vortex-lattice program.

Figure 3.1 compares the results from several programs and also presents an experimental data set which can be compared to the predicted total lift coefficients. The potential lift coefficient is of interest for comparing the theoretical analyses, but it cannot be separated from vortex lift in reality. From 0 to 12 degrees angle of attack, all three methods agree well with the experimental data, but APAS is slightly better; beyond 12 degrees, VLM-FIG is the best predictor. The agreement which APAS shows at these low angles of attack is, apparently, fortuitous.

APAS calculates aerodynamic coefficients for unit angles of attack and for unit flap deflections and then multiplies the coefficient by the actual angle of attack of flap deflection to obtain the actual coefficient (Ref. 25). For this reason, APAS results are completely dependent on the low angle of attack aerodynamic characteristics. The slope of the potential lift curve is theoretically greatest at $\alpha = 0°$ and gradually decreases with increasing α. If the slope at $\alpha = 1°$ is used for all angles of attack, the potential lift curve will be linear and will pre-

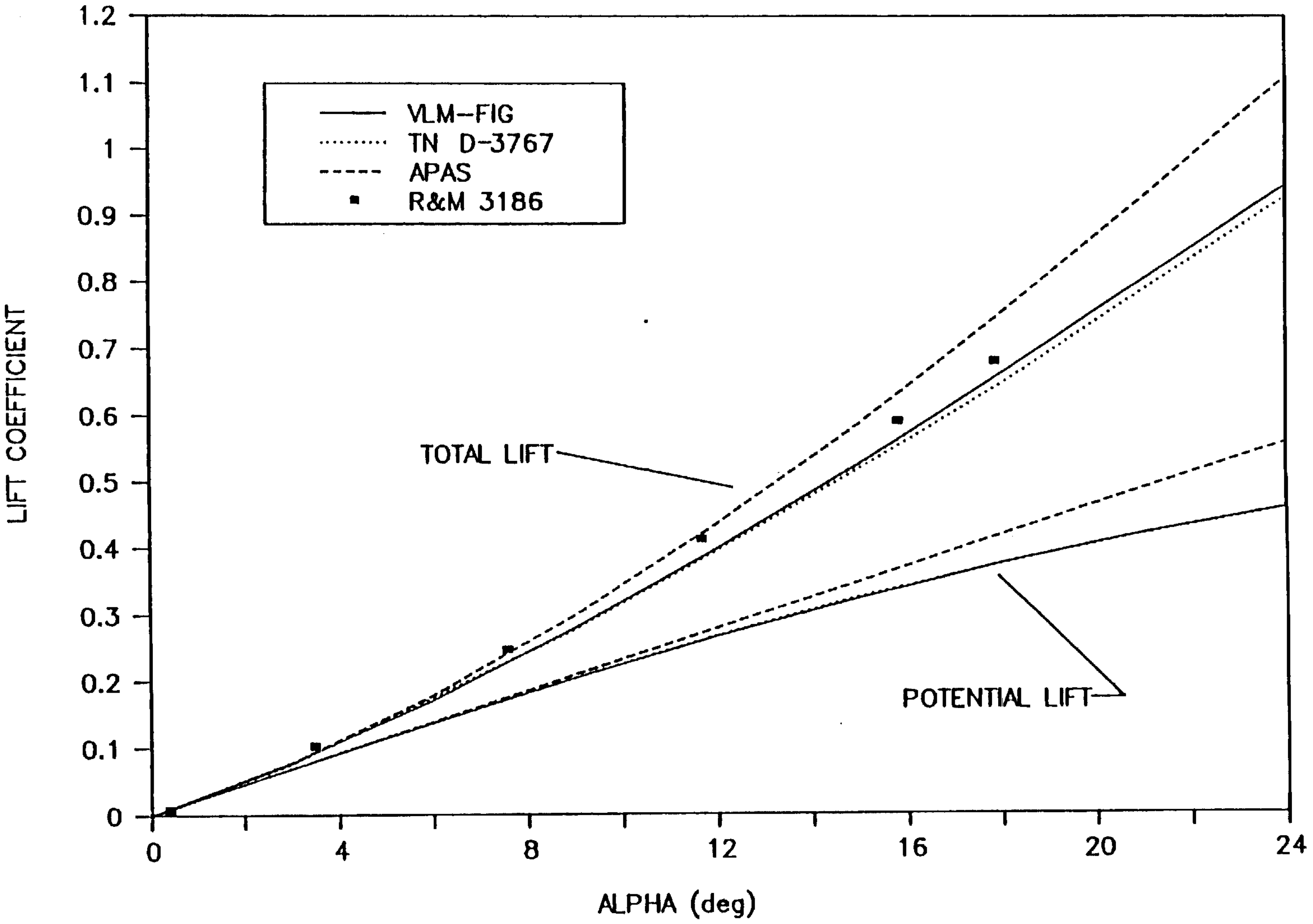

Figure 3.1 A Comparison of the Lift Coefficients Predicted by Different Theories and Experiment: 76° delta, OGE

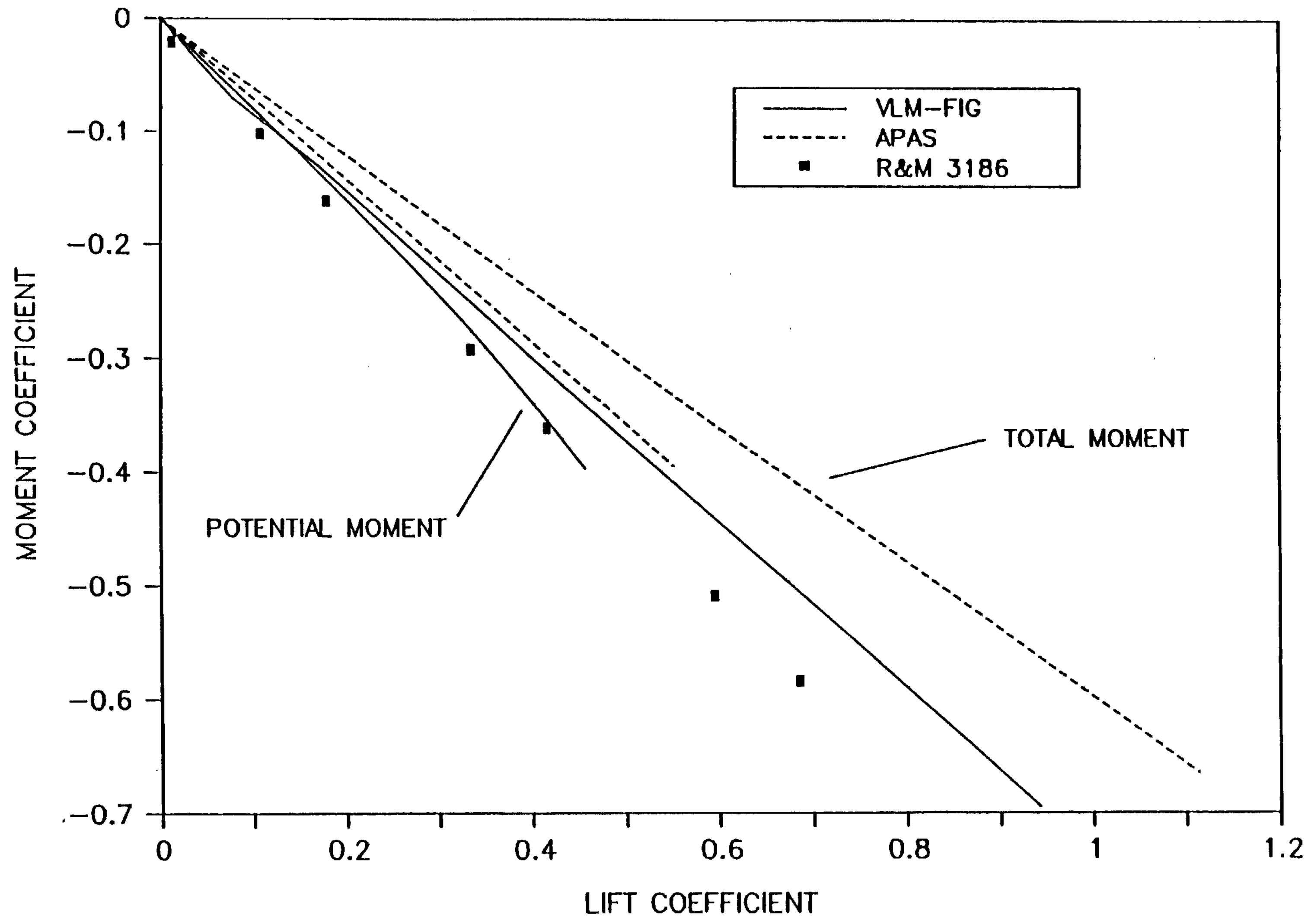

Figure 3.2 A Comparison of the Pitching Moment Coefficients Predicted by Different Theories and Experiment: 76° delta, OGE

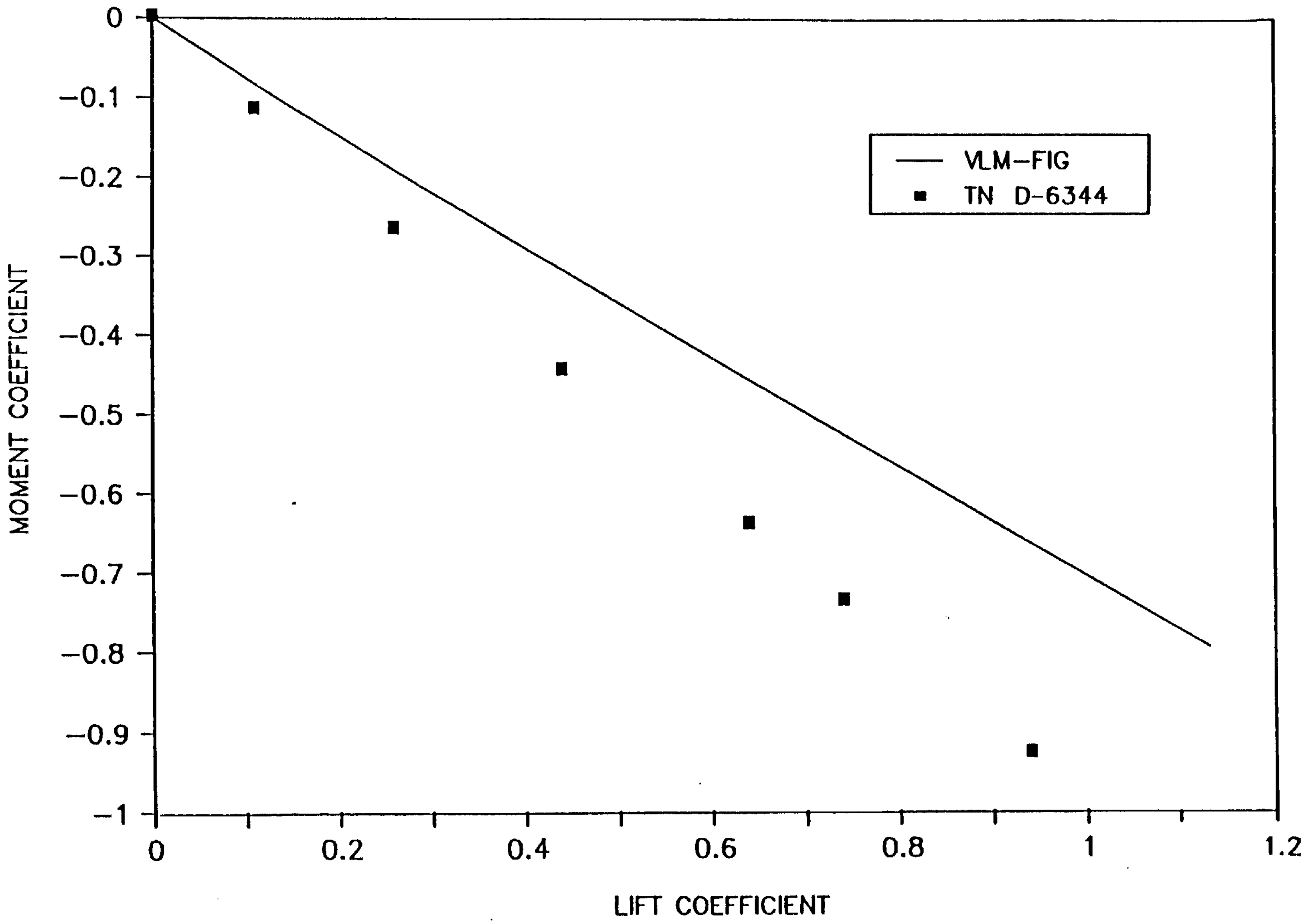

Figure 3.3 A Comparison of the Pitching Moment Coefficients Predicted by Theory and Experiment: 74° swept, 37° recessed arrow, OGE

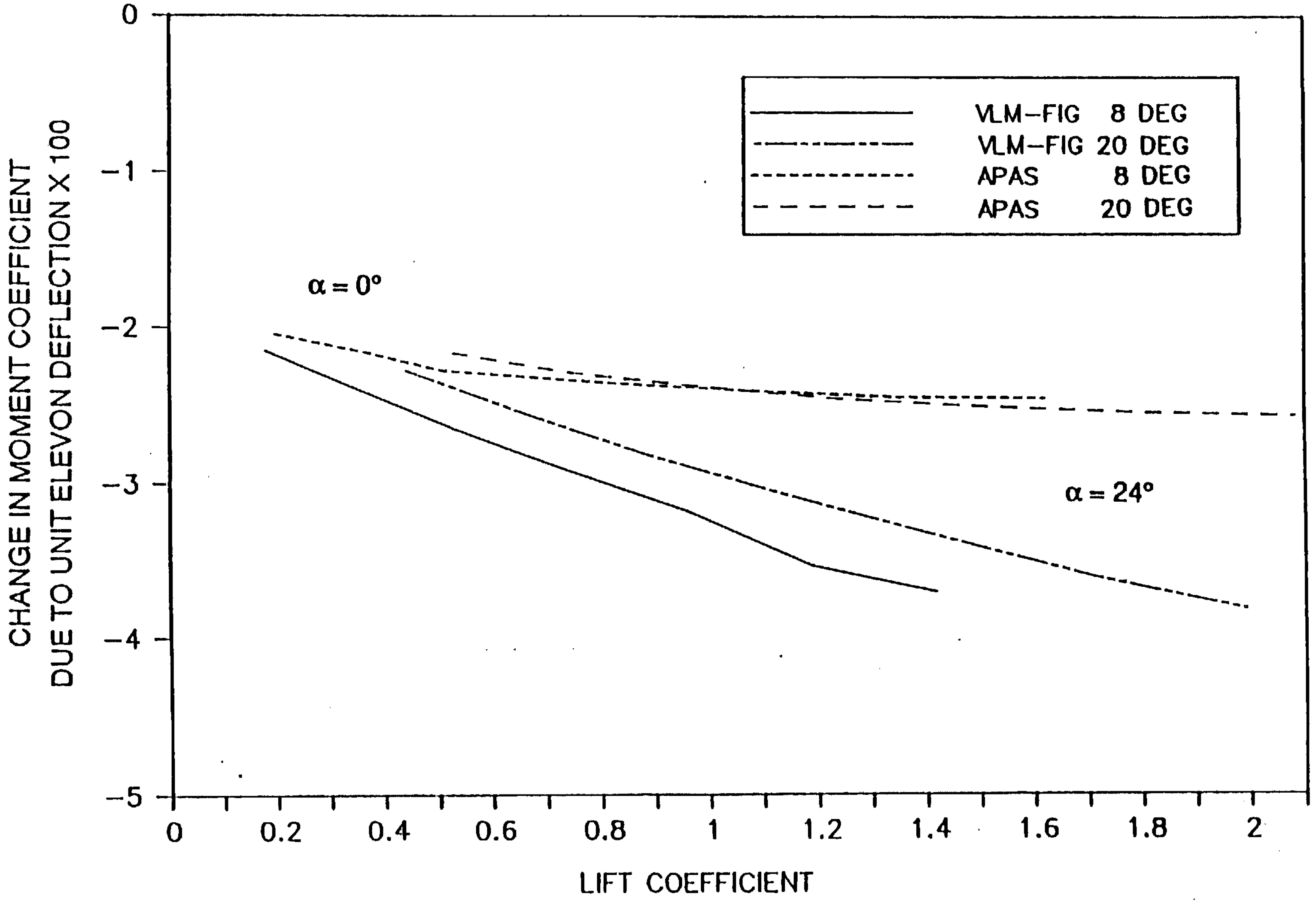

Figure 3.4 A Comparison of the Change in Pitching Moment Coefficient Due to Unit Elevon Deflections as Predicted by Two Theories: 70° delta, δ_c=8° and 20°, OGE

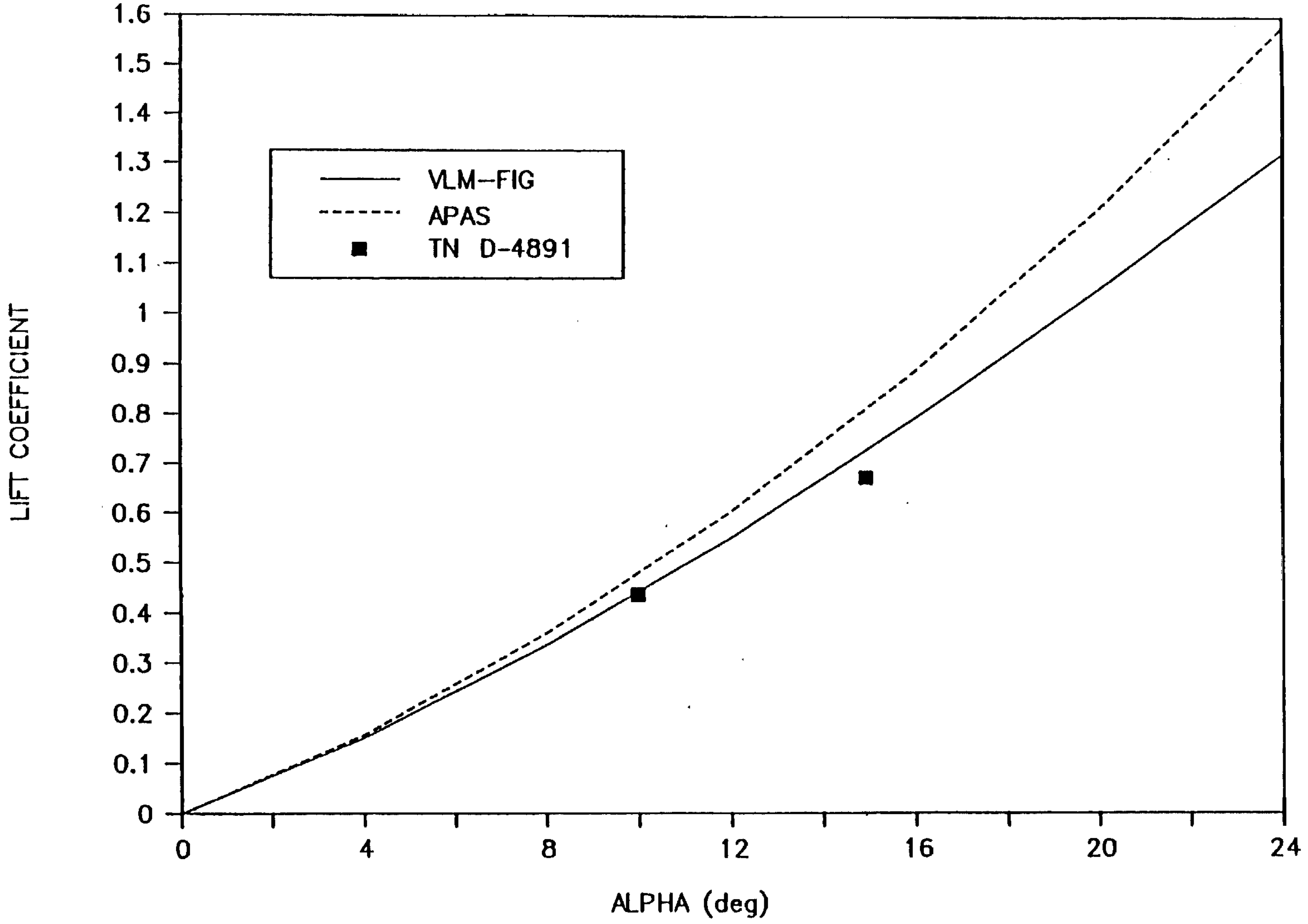

Figure 3.5 A Comparison of the Lift Coefficient Predicted by Theory and Experiment: 70° delta, h/b=.465

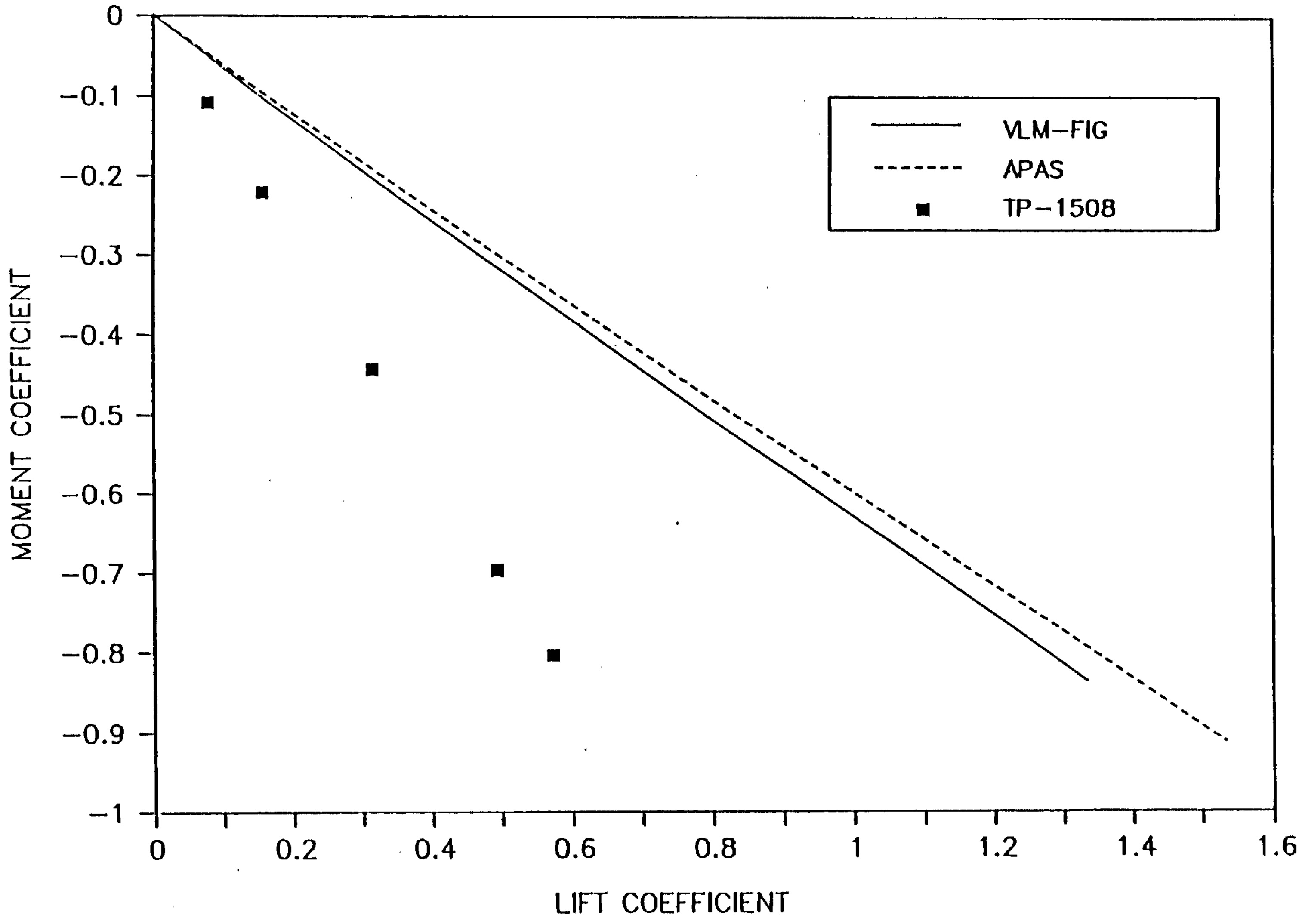

Figure 3.6 A Comparison of the Pitching Moment Coefficients Predicted by Theory and Experiment: 70° arrow, h/b=.274 to .30 (theory); 70° arrow configuration, h/b=.30 (experiment)

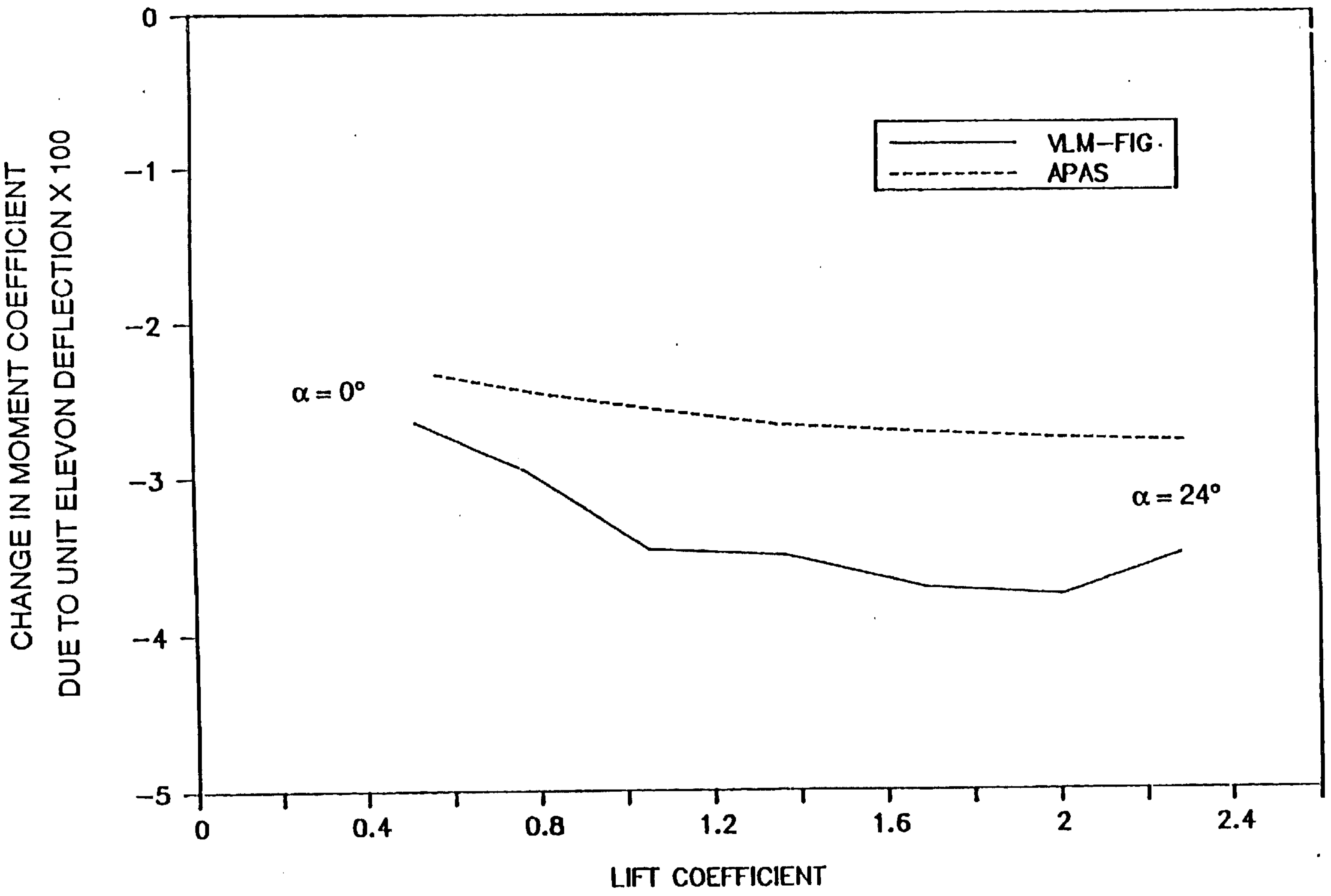

Figure 3.7 A Comparison of the Change in Pitching Moment Coefficient Due to a Unit Elevon Deflection as Predicted by Theory: 70° delta, δ_e=20°, h/b=.30

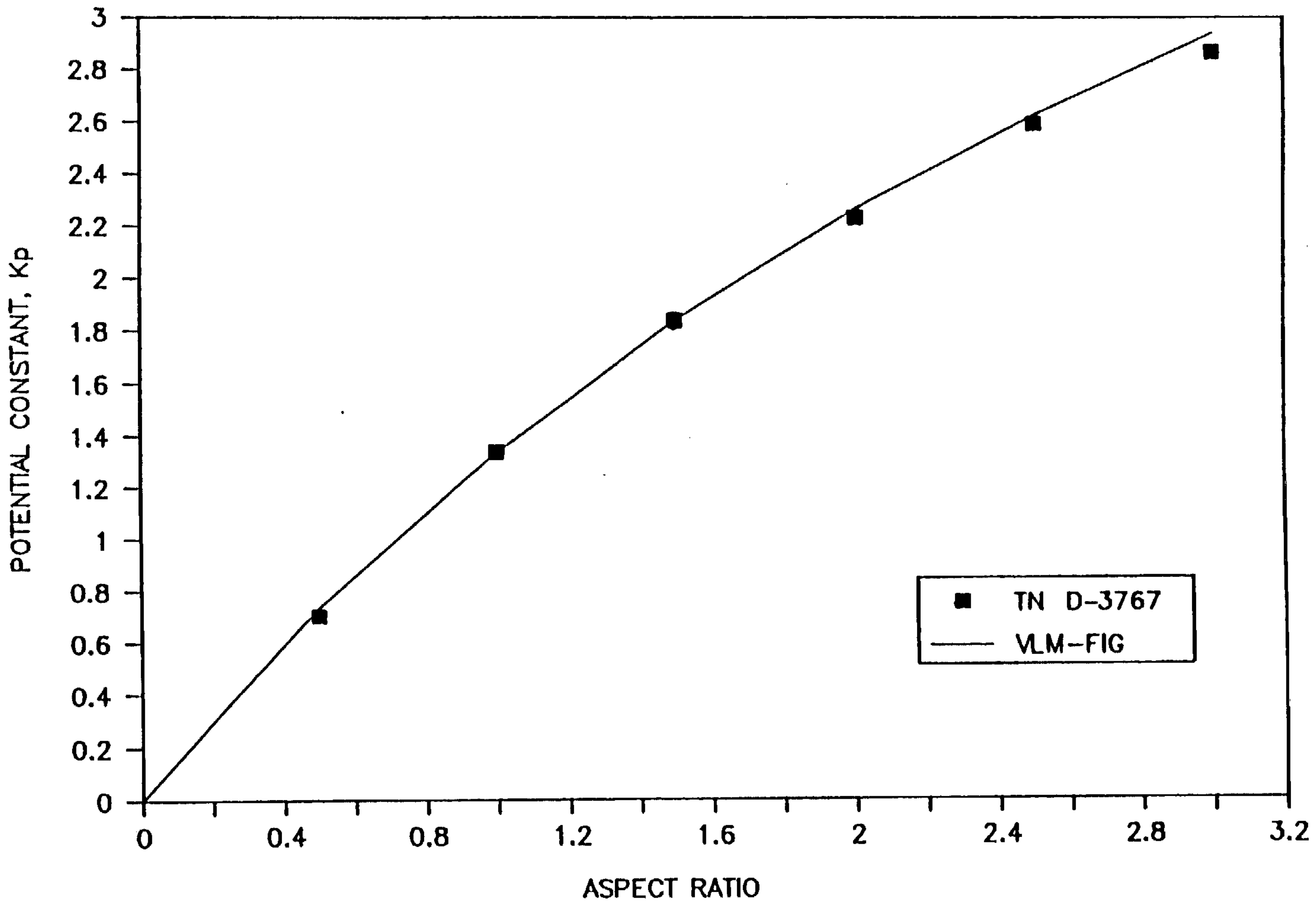

Figure 3.8 A Comparison of the Potential Constant for Delta Wings with Various Aspect Ratios: OGE

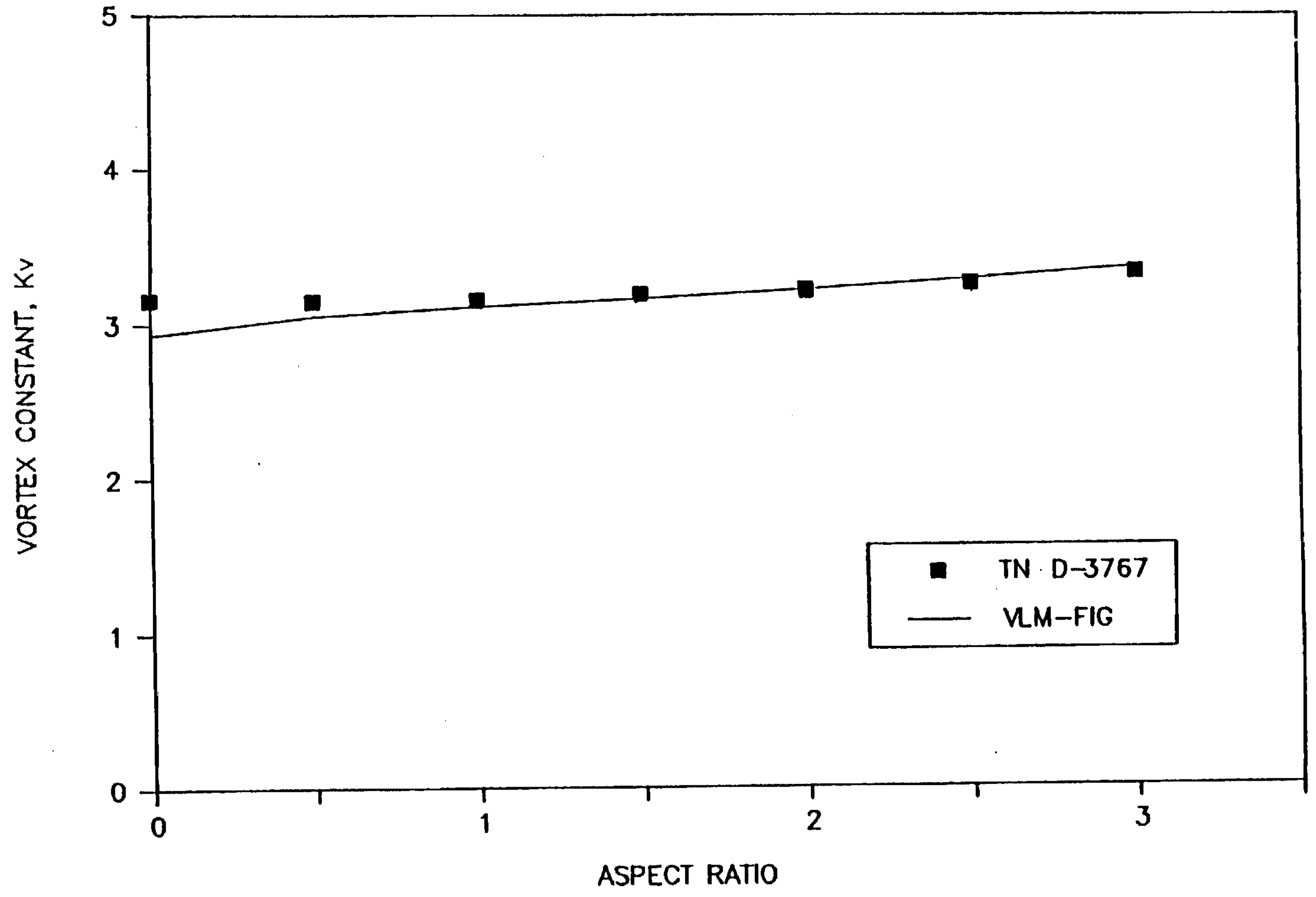

Figure 3.9 A Comparison of the Vortex Constant for Delta Wings with Various Aspect Ratios: OGE

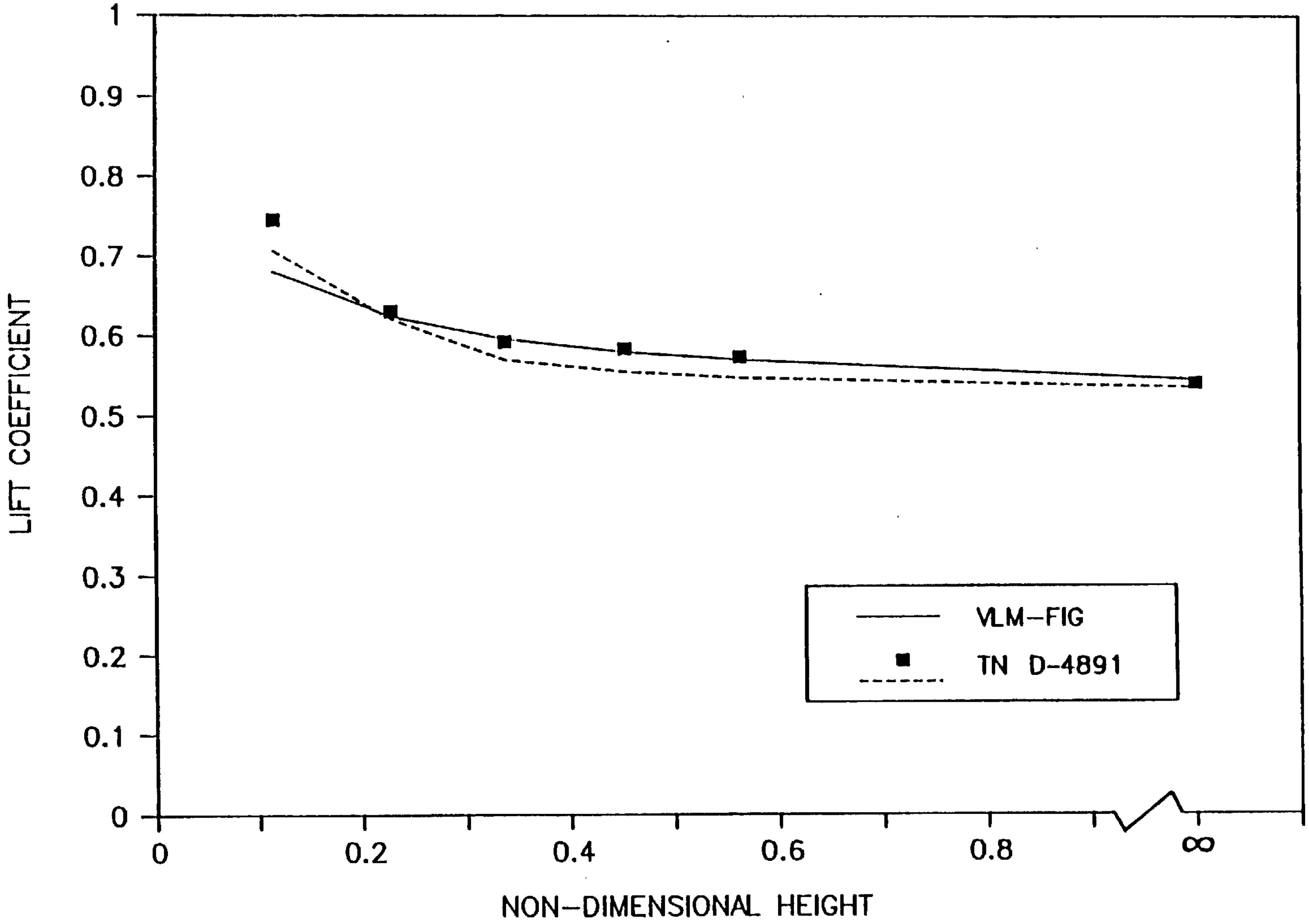

Figure 3.10 A Comparison of the Effect of Ground Proximity on the Lift Coefficient Predicted by Theory and Experiment: 75° delta, α=10°

dict potential lift coefficients which exceed those predicted by VLM-FIG type programs. This scenario matches exactly the potential lift curve in Figure 3.1. The vortex lift coefficient is directly proportional to a function of the slope of the potential lift curve so, if the slope is overpredicted, then the vortex lift is also overpredicted. Thus, APAS incorrectly overpredicts the potential lift, the vortex lift, and consequently, the total lift, resulting in apparently fortuitous agreement with the experiental results.

It is interesting to note that VLM-FIG and TN D-3767 predict identically the potential lift curve throughout the angle of attack range, but they predict slightly different total lift coefficients at higher angles of attack. Since VLM-FIG uses a relaxed-wake approach and TN D-3767 does not, it can be inferred that the effect of the relaxed wake out of ground effect is to alter the vortex lift coefficient only. This is equivalent to stating that the free wake causes the vortex constant, K_v, to become a function of the angle of attack, but that it does not strongly affect the potential constant, K_p.

Figure 3.2 compares the pitching moment coefficients predicted by VLM-FIG with those predicted by APAS and experimental data. VLM-FIG produces results which eliminate about 60% of the error which would be obtained using APAS. The longitudinal potential lift and vortex lift loadings are similar, but the centroid of the potential lift is located somewhat farther aft than the vortex lift, except at low lift coefficients. Thus, the potential lift moment is greater than the vortex lift moment for all but the lowest lift coefficients. Near $C_L = .11$ the two are about equal. In this figure, the vortex moment is not represented by the difference between the total and potential moments since the potential moments are plotted versus potential lift coefficient and the total moments are plotted versus total lift coefficients.

The change in slope of the total moment coefficient curve at $C_L = .08$ is most likely due to onset of measurable amounts of vortex lift. This is shown in Figure 3.1 as a change in the slope of the total lift coefficient at $\alpha = 3°$ and such a change would have the effect of shifting the total moment curve to the right, as in Figure 3.2.

Figure 3.3 is intended to illustrate that the moment coefficient is not well predicted by the suction analogy if the wing has an arrow type planform, as verified experimentally in TN D-6344. An arrow planform can be considered a delta planform with additional outboard wing area behind the trailing edge. This additional area allows the leading edge vortex to persist further aft and create a stronger nose-down moment than that predicted by the suction analogy. Other figures which display comparisons to arrow wings should be viewed in this light (Ref. 26).

Figure 3.4 shows the change in the moment coefficient due to unit elevon deflection centered at 8° and 20°. Both APAS and VLM-FIG show that the nose-down moment which is generated will increase as the lift coefficient increases, but the APAS values do not change as much with lift coefficient. VLM-FIG predicts decreasing control power for higher control deflections, while APAS predicts that control power is nearly independent of the control deflection. This difference probably results from the free-wake analysis of VLM-FIG, which is able to model the non-linear effects at high deflection angles. The variations observed in the APAS predictions result from non-linearities inherent to the suction analogy.

Figure 3.5 compares the predictions generated by VLM-FIG, APAS, and experimental data when a 70° delta wing enters ground effect. Again, VLM-FIG produces better results than APAS, although it still overpredicts the experimental value obtained at $\alpha = 15°$. At such intermediate angles of attack the leading-edge vortex begins to migrate upward at the rear of the wing due to real fluid effects. This causes a decrease in lift and pitching moment coefficients which could explain the theoretical overprediction.

There is no standardized procedure for non-dimensionalizing the height of the wing above the ground. For this investigation, the height above the ground is defined as the height of the root-chord trailing edge and it is non-dimensionalized by the wingspan. Fox (Ref. 19), for example, defined the height as the height of the local quarter-chord point of the mean aerodynamic chord and non-dimensionalized by the mean aerodynamic chord. Thus, conve rsions were necessary to compare heights which had been non-dimensionalized differently. Unfortunately, the wing height above the ground is a function of the angle of attack of the wing which disallows a one-to-one height conversion factor from the reference data set to the VLM-FIG data set. Each height, therefore, requires its own angle of attack conversion. Using Figure 3.5, as an example, the reference's non-dimensional height was .4, which corresponded to a VLM-FIG non-dimensional height of .4625 at $\alpha = 10°$ and .4675 and $\alpha = -15°$. In this case, the conversion is almost independent of angle of attack and there is almost a one-to-one conversion. The height differences for other figures are much greater, however, and cannot be ignored. The range of non-dimensional heights used will be stated in each figure, as needed.

Figure 3.6 shows how the ground affects the moment coefficient of a 70° arrow type planform. This planform has the same leading-edge sweep angle and notch ratio of a more complex planform, which was part of a configuration, presented in TP-1508. Based on this idealization and the intrinsically poor arrow wing prediction capabilities, the figure is internally consistent.

The change in moment coefficient due to a 1° change in elevon position at a deflection of 20° on a 70° delta wing is shown in Figure 3.7. As in Figure 3.4, VLM-FIG predicts a greater nose-down moment than APAS. When Figures 3.4 and 3.7 are compared, it becomes evident that both methods are affected by the ground in a similar way, namely, that the ground provides additional control power. The increment predicted by VLM-FIG is about that which APAS predicts for low to medium lift coefficients. At a lift coefficient of 1.0, however, the kink in the

VLM-FIG moment coefficient causes a cross-over with the values predicted by VLM-FIG in Figure 3.4 and consequently, there is slightly less control power predicted at a lift coefficient of 2.0. APAS shows no such change, but rather a continuously increasing difference between the two as the lift coefficient increases.

Figures 3.8 and 3.9 compare the values of the potential and vortex constants as predicted by two methods. For an aspect ratio of 1.0, the two methods match, which causes the potential lift agreement found in Figure 3.1. The vortex lift constant, which becomes a function of angle of attack when the wake is relaxed, is calculated at low angle of attack for generating the data in Figue 3.9. At low angles of attack, VLM-FIG predicts a vortex constant which is just less than that predicted by TN D-3767. At high angles of attack, however, VLM-FIG predicts a vortex constant which is greater than that predicted by TN D-3767; this is evident in Figure 3.1 as a cross-over near 8° of the values predicted by these two methods.

Figure 3.10 compares two theories and an experiment and shows that lift increases due to ground effect as the wing approaches the ground. Both closely predict the experimental values, but VLM-FIG is better until a non-dimensional height of about 0.2. The difference between the two predictive methods is primarily that VLM-FIG has a relaxed wake and TN D-4891 does not. For this reason, it seems strange that TN D-4891 apparently predicts the lift better at very low heights above the ground. As with Figure 3.7 , a non-dimensional height of 0.11

represents a true height of $.05c_o$, which is so close to the ground that the airplane will most likely have landed.

VLM-FIG Predictions

The nose-down moment coefficient increases near the ground as shown in Figure 3.11. The coefficient $\partial C_M/\partial\alpha$ is nearly constant with changing non-dimensional height, portrayed by the nearly constant difference between the two curves in the figure. At lower non-dimensional heights $\partial C_M/\partial\alpha$ is slightly greater, indicating that as the airplane begins to flare and increase its angle of attack, the nose-down pitching moment increases slightly. As with any airplane, it is important that this parameter does not markedly decrease as either α is increased or h/b is decreased, since this could lead to an unstable situation as the airplane descends. As such, however, the longitudinal stability does not decrease when the airplane is operating in ground effect.

Figure 3.12 was generated by using data obtained from VLM-FIG when it operated in its normal mode and when it was forced to terminate after the flat wake aerodynamic characteristics had been calculated. It shows that most of the out-of-ground-effect discrepancies between the two methods occur at higher angles of attack, which is consistent with the limitations of linearized theory. At an angle of attack of 24°, the ratio of the change in vortex lift to the change in potential lift, $\Delta C_{Lv}/\Delta C_{Lp}$, is 3.7. This supports the assertion made concerning Figure 3.1 that the effect of the free wake is to alter primarily vortex lift.

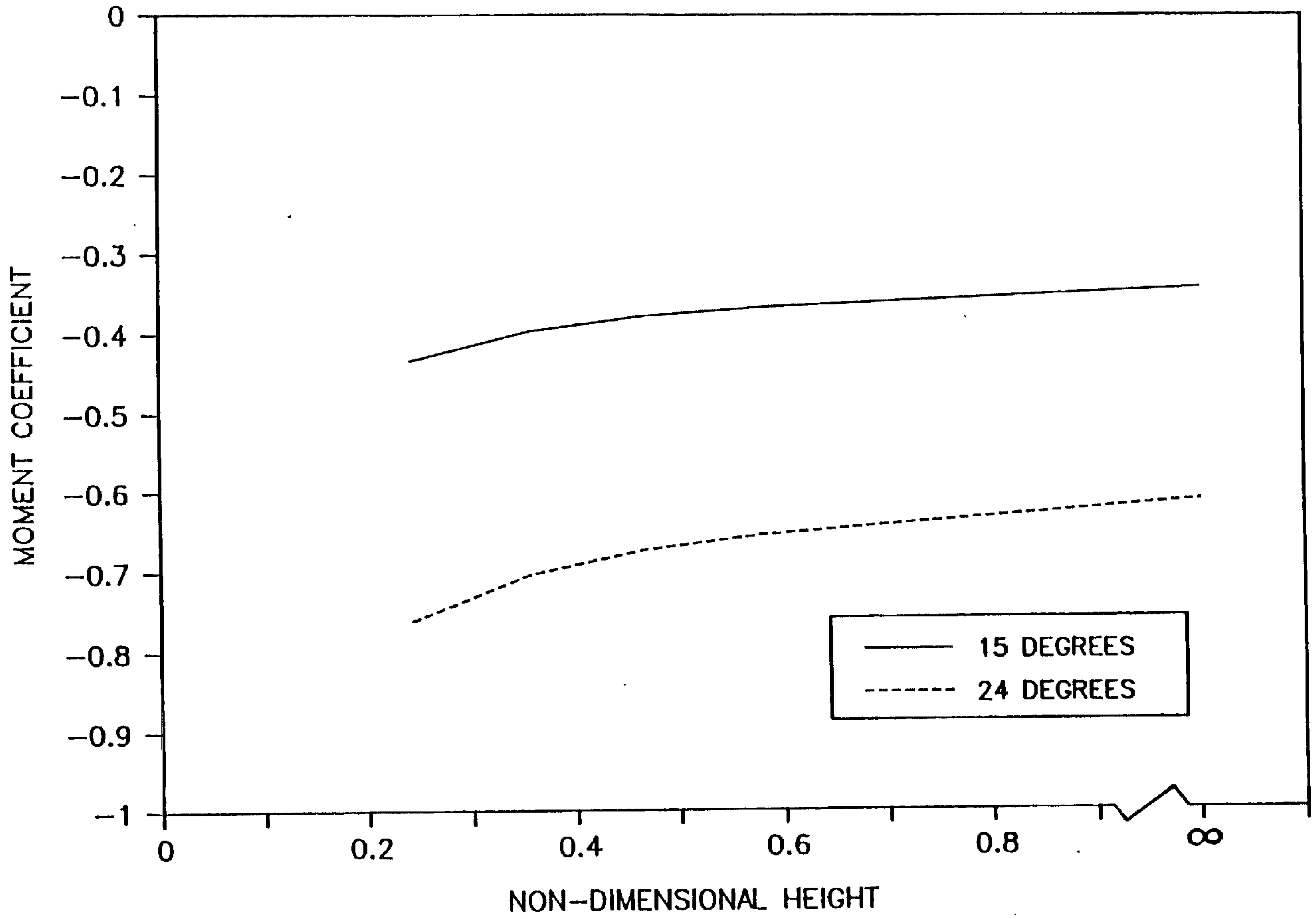

Figure 3.11 A Comparison of the Effect of Ground Proximity on the Pitching Moment Coefficient Predicted by VLM-FIG: 70° delta, α=15° and 24°

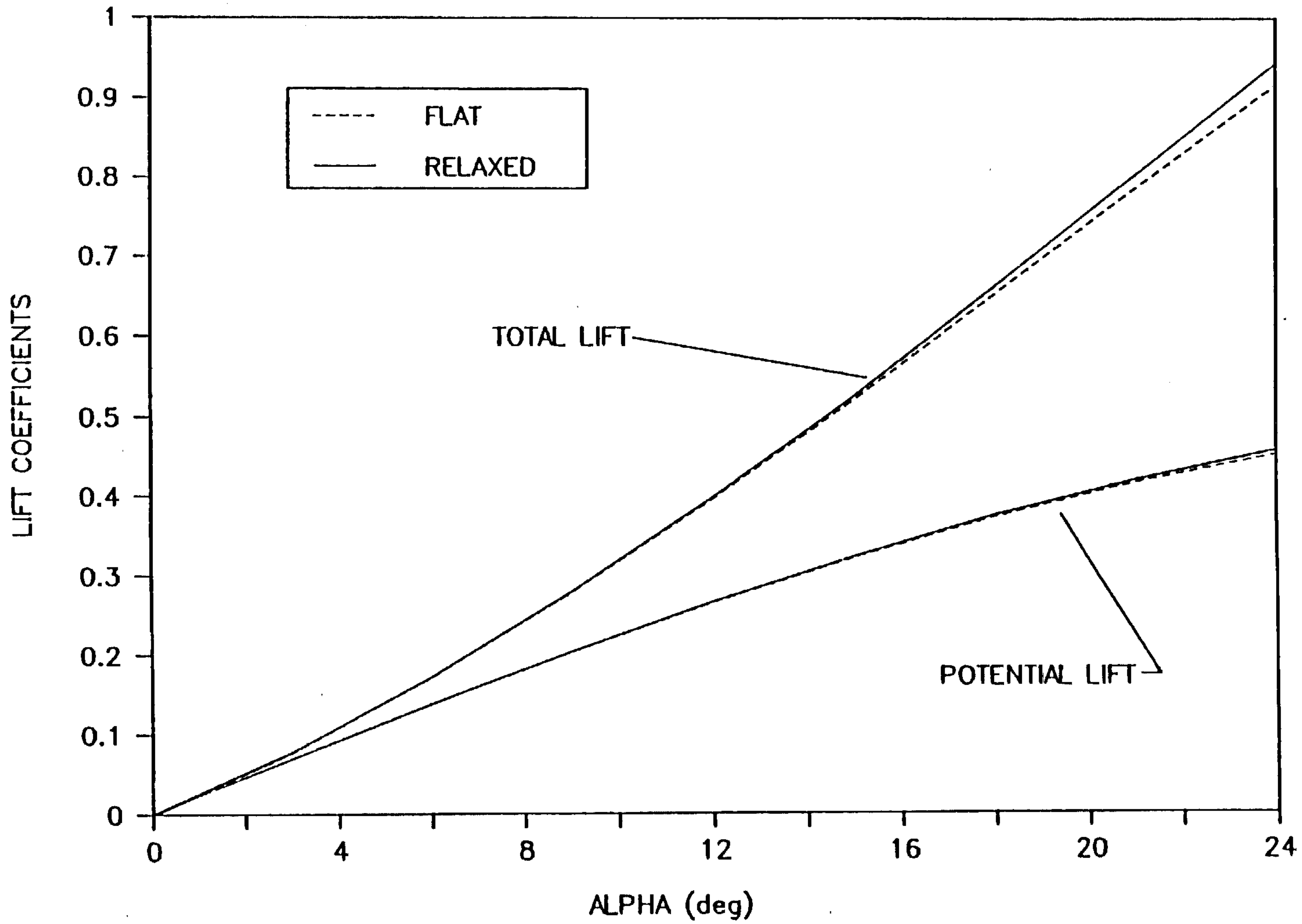

Figure 3.12 A Comparison of the Lift Coefficient Predicted by VLM-FIG with a Flat Wake and a Relaxed Wake: OGE

Figure 3.13 also compares the flat and relaxed wake values which were generated by VLM-FIG. The potential and total moments for both wakes are identical, indicating that the centroid of the lift does not change when the wake is relaxed, even though the lift does change. An interesting feature of this figure is that the slope of the potential moment becomes more negative with increasing lift coefficient but the total moment slope remains constant with lift coefficient. The significance of this feature is unclear at this time, but it implies that a non-linear vortex moment adds to the potential moment in such a way to keep the total moment a constant function of the lift coefficient.

The differences between the flat and relaxed wake predictions are much different when a wing is analyzed in ground effect, as Figure 3.14 illustrates. At low angles of attack, the potential lift and vortex lift are higher for the flat wake method than for the relaxed wake method. According to Figure 3.15, the percent change between the lift coefficient in ground effect and out of ground effect is greatest for the flat wake near an angle of attack of 3°. This is reflected by the decreased slope of the flat wake curves in Figure 3.14. The reason for these changes in slope is that until the angle of attack reaches about 3°, the increased circulation is more strongly influenced by the height of the wing above the ground, but after that, is more strongly influenced by the angle of attack of the wing. For the relaxed wake, it seems that most of the increase in lift coefficient occurs due to ground proximity; however, there is some increase in the lift coefficient due to

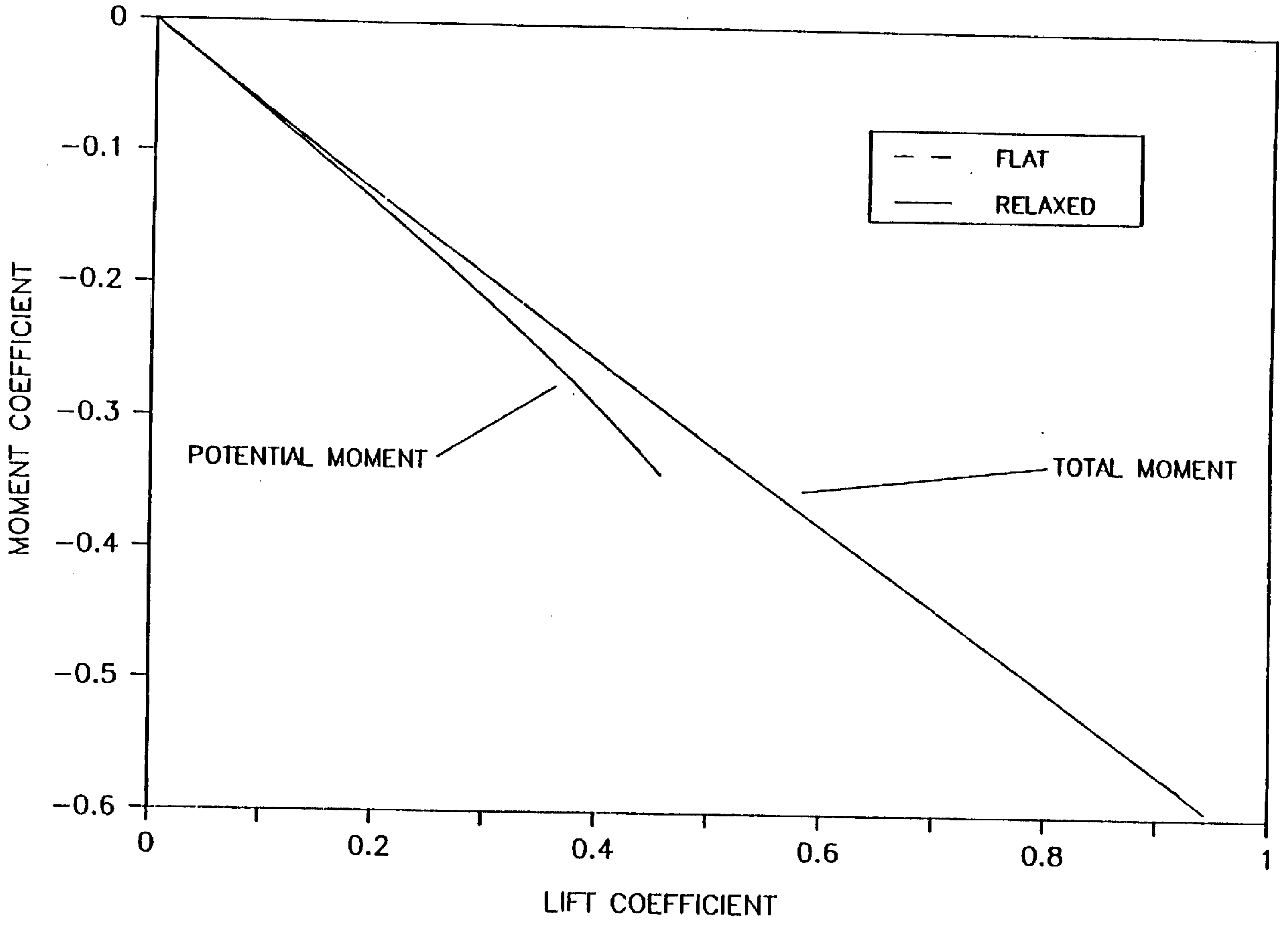

Figure 3.13 A Comparison of the Moment Coefficients Predicted by VLM-FIG with a Flat Wake and a Relaxed Wake: OGE

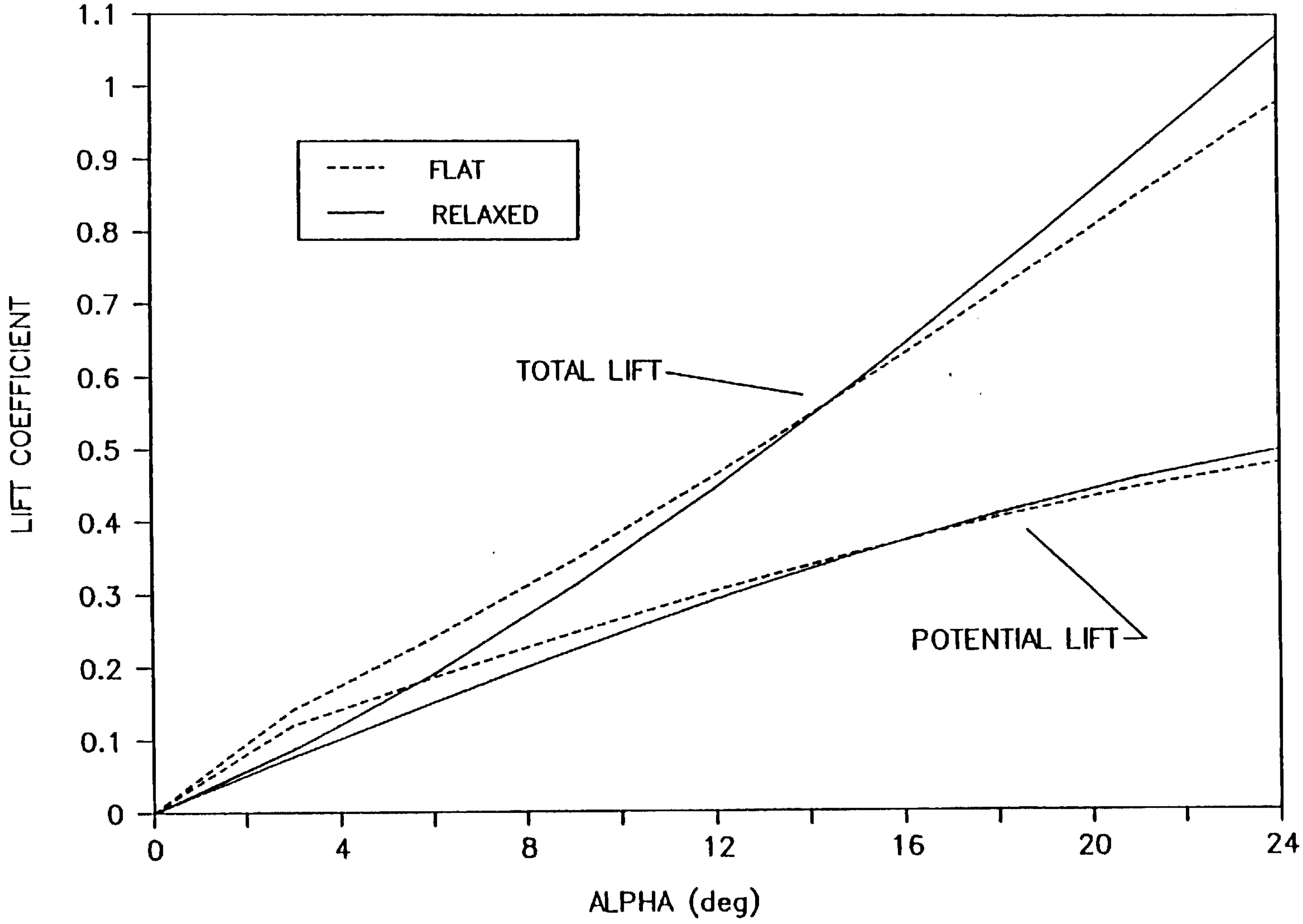

Figure 3.14 A Comparison of the Lift Coefficients Predicted by VLM-FIG with a Flat Wake and a Relaxed Wake: h/b=.4

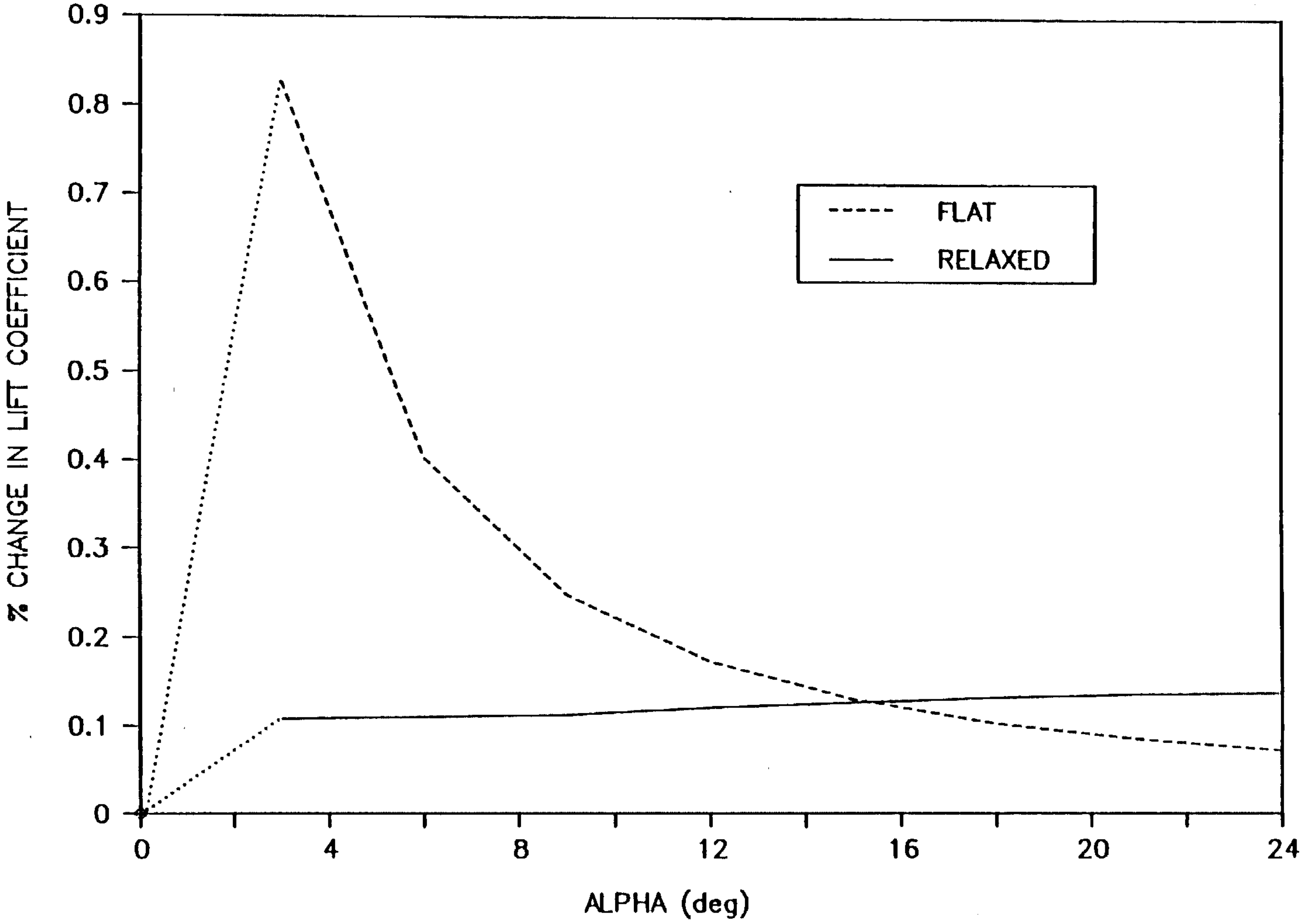

Figure 3.15 VLM-FIG Prediction of the Percent Change in Lift Coefficient Due to Ground Effect for a Flat Wake and a Relaxed Wake: 76° delta, h/b=.4

operating at high angles of attack near the ground. These curves agree with the general trends and orders of magnitude of data presented in Reference 4.

Figure 3.16 shows that the relaxed wake method predicts a pitching moment which differs by a few percent from that predicted by the flat wake method when operating in ground effect. Thus, the center of pressure moves slightly aft due to the relaxed wake in ground effect. This could be a result of the relaxed wake being more strongly constrained by its image wake than the flat wake is constrained by its flatness condition. It seems logical that the rear portions of the wing would then be more affected by this than the front portions, resulting in rearward center of pressure movement.

Figure 3.17 is a comparison of the lift coefficient in ground effect and out of ground effect as a function of the angle of attack. The curves were generated by combining the relaxed wake results of Figures 3.12 and 3.15. The increase in potential lift accounts of about 1/3 of the total lift increase at $\alpha = 24°$, while it accounts for almost all of the total lift increase at low angles of attack. Since the vortex and potential lifts are nearly equal at $\alpha = 24°$, and since the increase in the vortex lift is twice that of the potential lift, it can be inferred that the ground affects the vortex lift more than it affects the potential lift. At such high angles of attack, however, real-fluid effects cause the leading-edge vortices to move upward off the aft portions of the wing, causing a decrease in vortex lift. Therefore, it should be expected that, in reality, the ground will cause the vortex lift to rep-

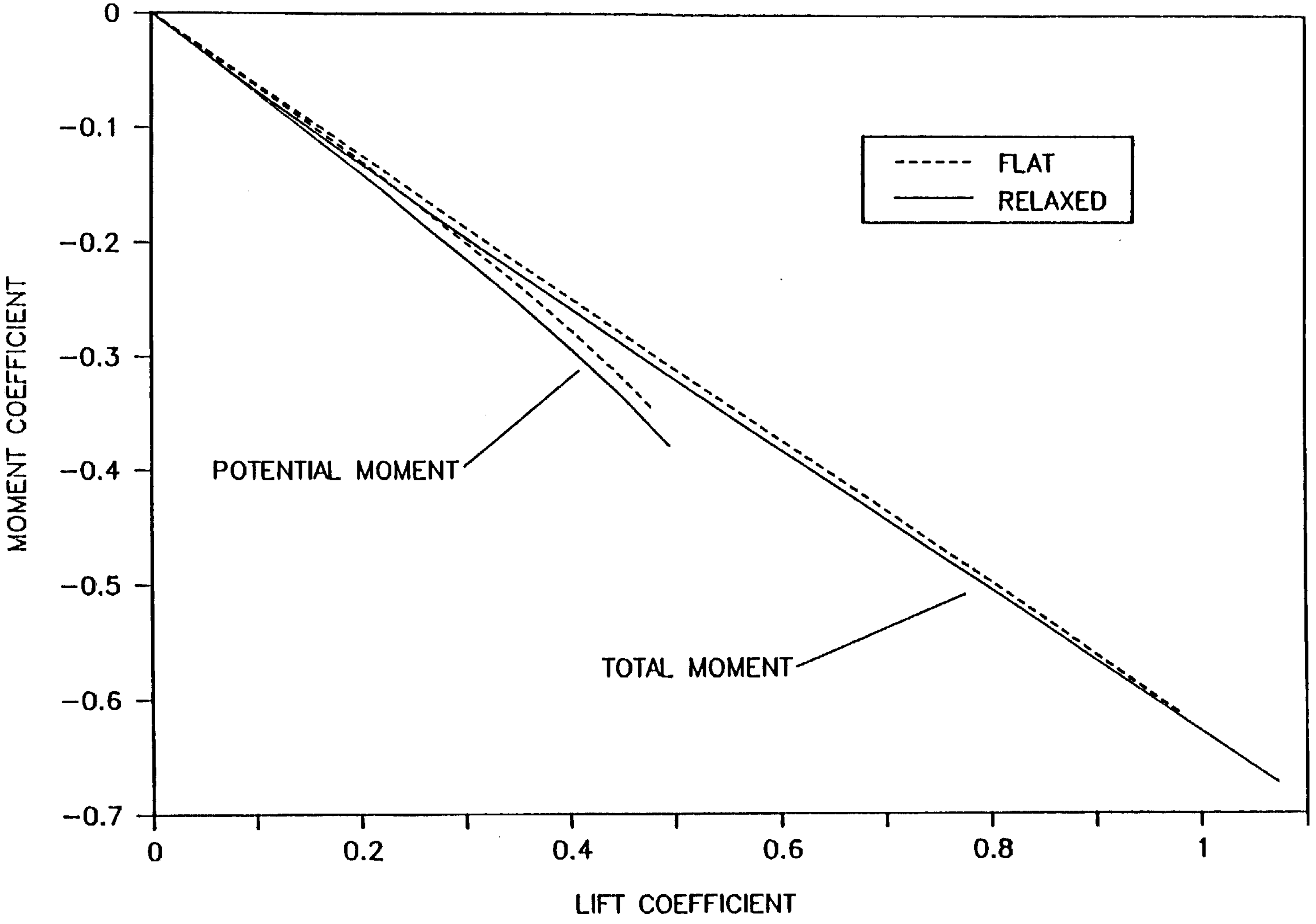

Figure 3.16 A Comparison of the Pitching Moment Coefficients Predicted by VLM-FIG with a Flat Wake and a Relaxed Wake: h/b=.4

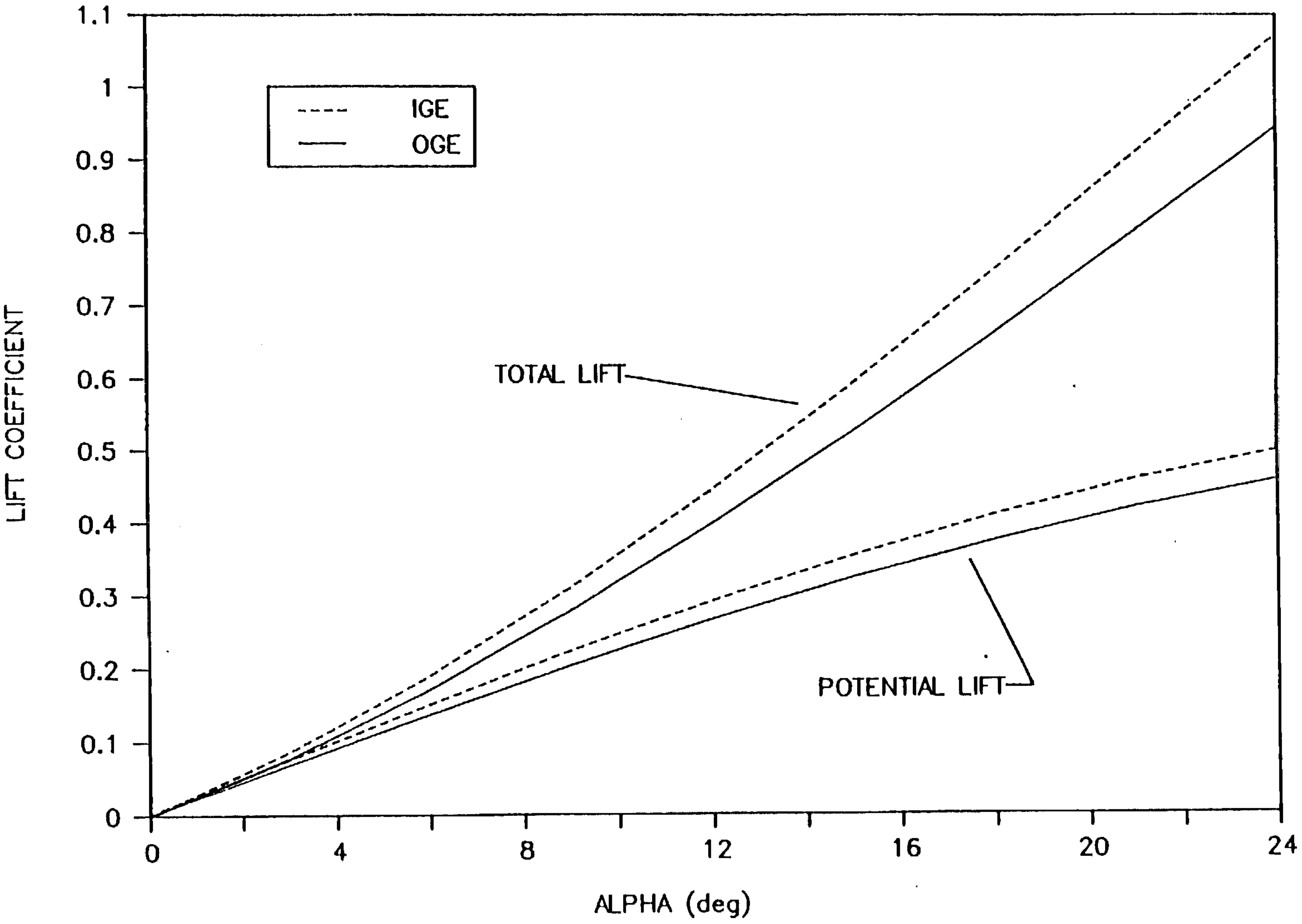

Figure 3.17 VLM-FIG Predictions of the Lift Coefficients In and Out of Ground Effect: 76°delta, h/b=.4

resent a smaller percentage of the total lift than VLM-FIG predicts. The effect that an image leading-edge vortex has on the wing leading-edge vortex should be modelled by VLM-FIG, since the vortex strength is determined in part by the downwash and since the downwash is included in the theory.

Figure 3.18 shows that the moment coefficient does not change significantly when the wing is in ground effect. The data were generated using the relaxed wake approach and show that the centroid of the lift is unaffected by the ground effect. This graph is a composite of the relaxed wake moment curves shown in Figures 3.13 and 3.16.

The change in the lift coefficient due to a unit elevon deflection centered at 8° as the wing approaches the ground is shown in Figure 3.19 for three angles of attack. As the wing nears the ground, a unit flap deflection results in a greater change in the lift coefficient. For α = 8° and 16°, the slope of the curves continuously increases, but for α = 24°, the change in the slope of the curve decreases when the wing reaches a non-dimensional height of 0.4. The cross-over also suggests that for low heights, there is an angle of attack which exploits both the angle of attack effects and the height effects in such a way that the change in lift due to a given control deflection is maximized. This angle appears to be between 16° and 24° from the information in the figure.

Figure 3.20 shows the change in the moment coefficient due to a unit elevon deflection as the wing approaches the ground at three different angles of attack.

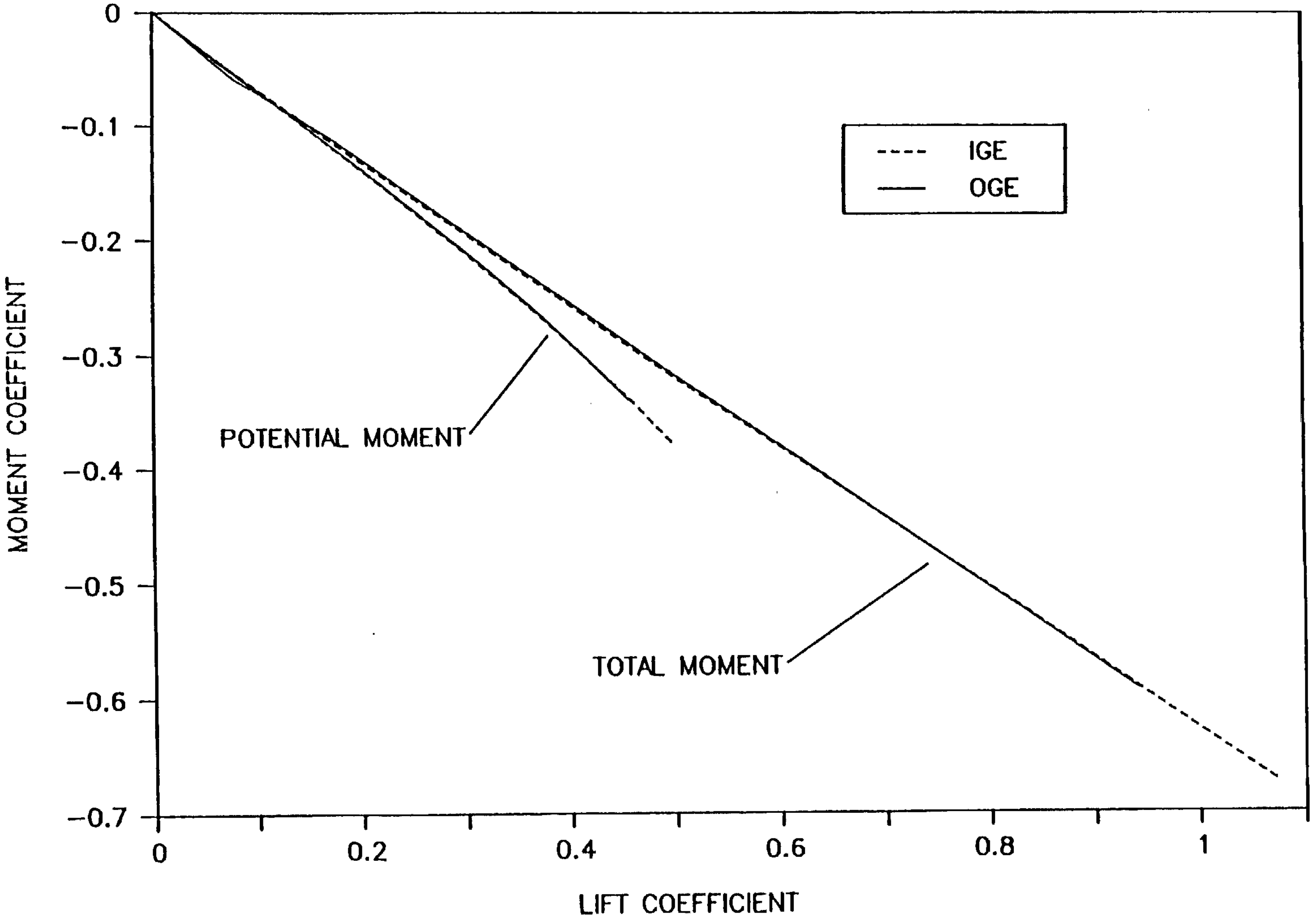

Figure 3.18 VLM-FIG Predictions for the Pitching Moment Coefficient In and Out of Ground Effect: 76° delta, h/b=.4

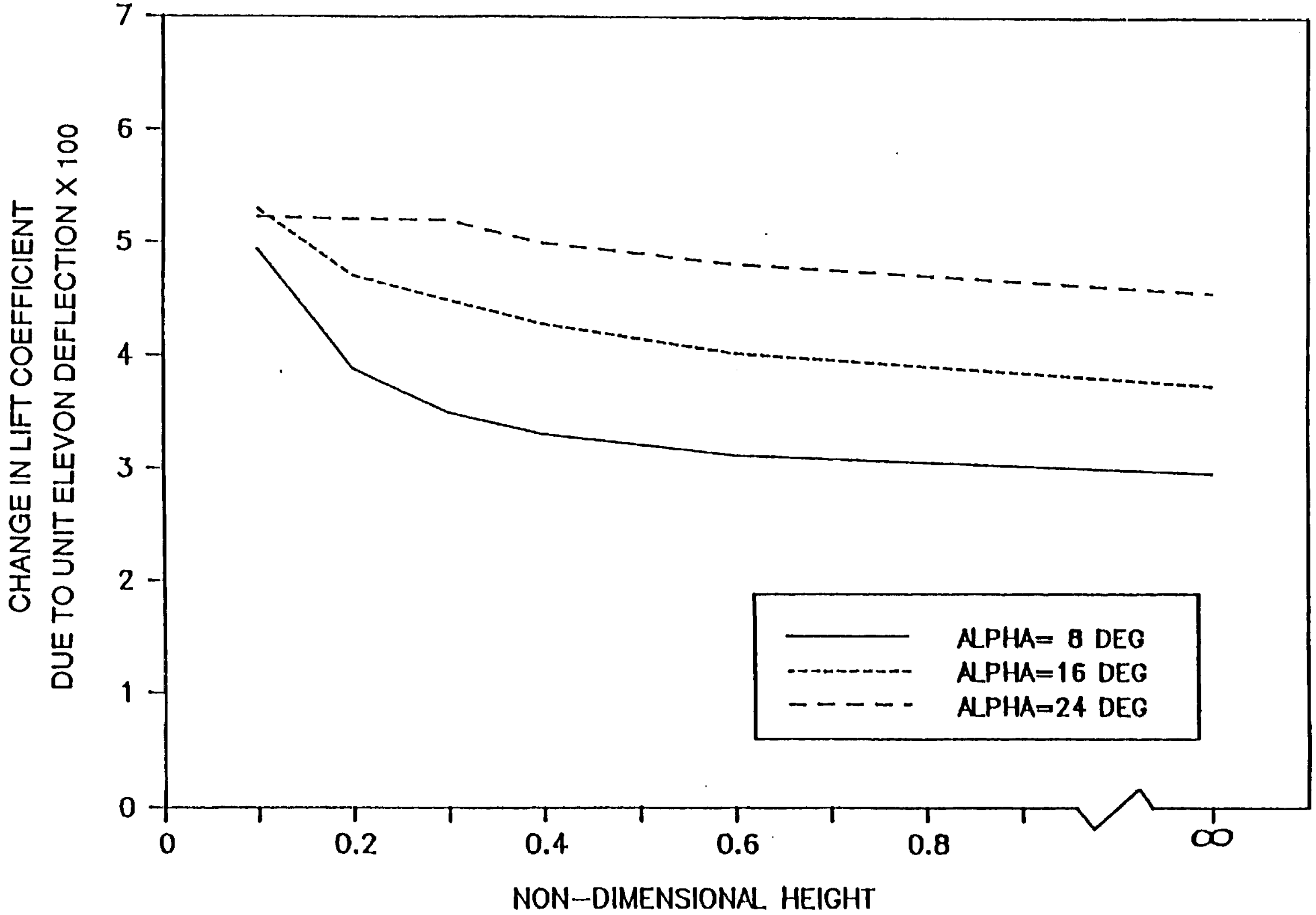

Figure 3.19 VLM-FIG Prediction of the Change in Lift Coefficient Due to a Unit Flap Deflection: 70° delta, δ_c=8.0°, IGE

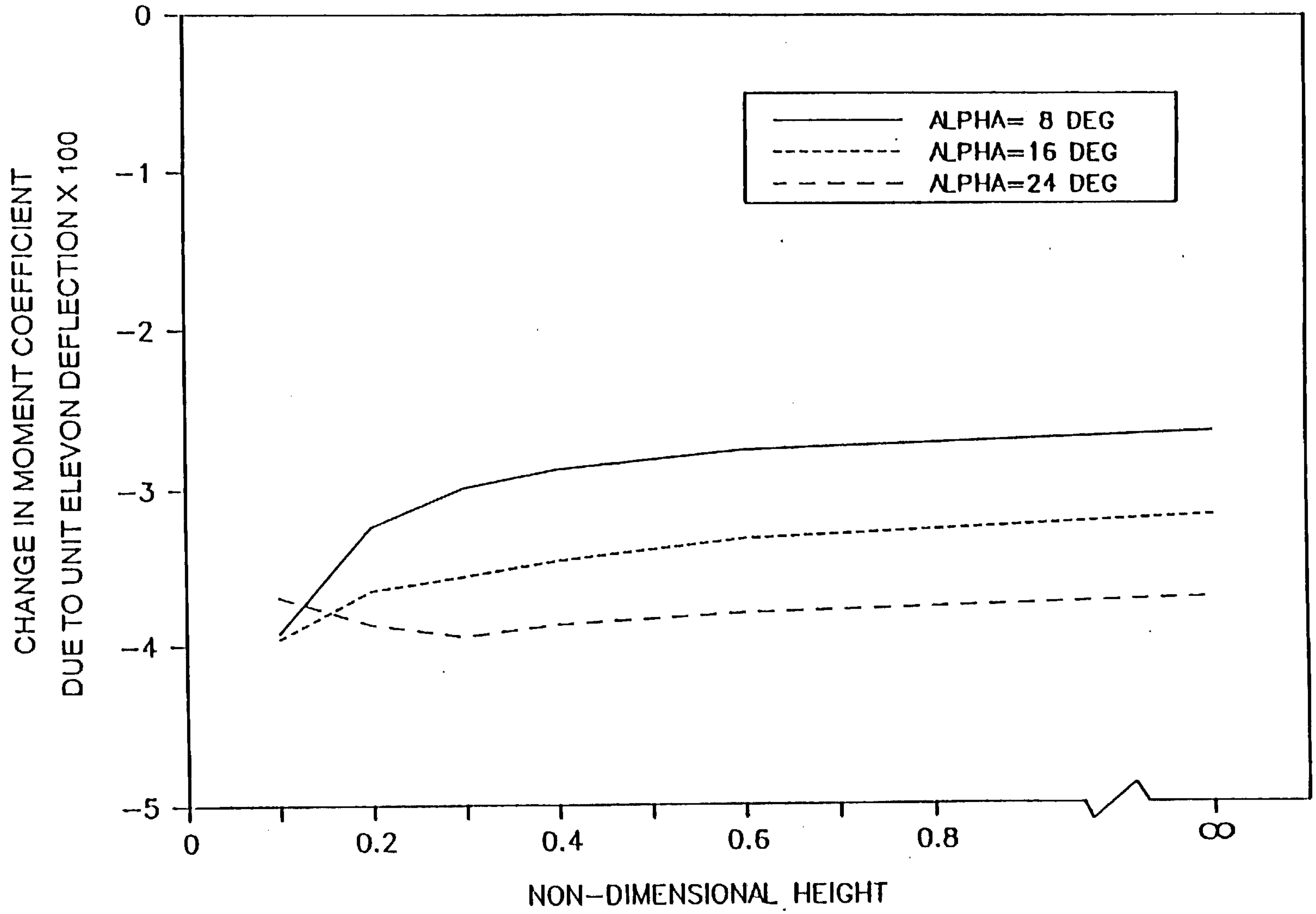

Figure 3.20 VLM-FIG Prediction of the Change in Pitching Moment Coefficient Due to a Unit Elevon Deflection: 70° delta, δ_e=8.0°, IGE

At low and medium angles of attack, the change in the nose-down moment increases as the non-dimensional height decreases. At high angles of attack, the nose-down moment increases until a non-dimensional height of 0.3, at which point the moment generated by a given control deflection decreases. Thus, there is a predicted slight decrease in control power for the combined conditions of a high angle of attack and a low height above the ground.

As noted for Figure 3.19, there appears to be an angle of attack between $\alpha = 16°$ and $\alpha = 24°$ which maximizes the control effectiveness about a given deflection at low heights above the ground. Simply, this means that both figures suggest that control power is not a monotonic function of the angle of attack when the wing is so close to the ground.

Finally, it should be pointed out that the changes in lift and moment coefficients displayed in Figures 3.19 and 3.20 were determined for a positive elevon deflection. A negative elevon deflection may have a somewhat different effect considering the possible interaction of the relaxed wake with the ground plane.

Summary and Conclusions

An investigation of the aerodynamic characteristics of highly swept delta wings with flaps operating in ground effect was conducted. A vortex-lattice computer program which incorporated a ground plane and a relaxed wake iteration scheme was developed to facilitate the research. The results generated by the program, VLM-FIG, were compared with experimental data and other similar programs to evaluate its ability to predict experimental results and its ability to improve upon previous programs, respectively. The following conclusions are presented:

1. It was found that VLM-FIG is a better predictor of aerodynamic characteristics than APAS, presumably because VLM-FIG uses a free-wake analysis.
2. VLM-FIG predicts that the moment due to a flap deflection in ground effect generally produces a greater increase in both C_L and C_M than the same deflection produces out of ground effect.
3. When results from VLM-FIG using a free and flat wake in and out of ground effect were analyzed, it was found that the ground effect and the free wake affect the vortex-lift characteristics of the wing more than the potential-lift characteristics. In addition, when the wing was evaluated in ground effect the effects of the free wake were significant.
4. VLM-FIG has been shown to be an effective tool for predicting stability and control derivatives. These derivatives are necessary for future work directed

towards a full static and dynamic stability and control analysis.

References

1. Polhamus, Edward C., "A Concept of the Vortex Lift of Sharp-Edge Delta Wings Based on a Leading-Edge-Suction Analogy," NASA TN D-3767, 1966.

2. Polhamus, Edward C., "Application of the Leading-Edge-Suction Analogy of Vortex Lift to the Drag Due to Lift of Sharp-Edge Delta Wings," NASA TN D-4739, 1968.

3. Polhamus, Edward C., "Predictions of Vortex-Lift Characteristics by a Leading-Edge Suction Analogy," *Journal of Aircraft*, Vol. 8, No. 4, April 1971, pp. 193–199.

4. Chang, R.C. and Muirhead, V.U., "Effect of Sink Rate on Ground Effect of Low-Aspect-Ratio Wings," *Journal of Aircraft*, Vol. 24, No. 3, March 1987, pp. 176–180.

5. Scott, W.B., "Rockwell's Simulator Emulates NASP Flight Characteristics," *Aviation Week and Space Technology*, October 23, 1989.

6. Snyder, M.H. and Lamar, J.E., "Application of the Leading-Edge-Suction Analogy to Prediction of Longitudinal Load Distribution and Pitching Moment for Sharp-Edged Delta Wings," NASA TN D-6994, October 1972.

7. Fox, C.H. and Lamar, J.E., " Theoretical and Experimental Longitudinal Aerodynamic Characteristics of an Aspect Ratio .25 Sharp-Edged Delta Wing at Subsonic, Supersonic, and Hypersonic Speeds," NASA TN D-7651, August 1974.

8. Lamar, J.E. and Glass, B.B., "Subsonic Aerodynamic Characteristics of Interacting Lifting Surfaces with Separated Flow Around Sharp Edges Predicted by a Vortex Lattice Method," NASA TN D-7921, 1975.

9. Lamar, J.E., "Minimum Trim Drag Design for Interfering Lifting Surfaces Using Vortex-Lattice Methodology," *Vortex Lattice Utilization*, NASA SP-405, 1976.

10. Pittman, J.L. and Dillon, J.L., "Vortex Lattice Prediction of Subsonic Aerodynamics of Hypersonic Vehicle Concepts," *Journal of Aircraft*, Vol. 14, No. 10, October 1977, pp. 1017–1018.

11. Bradley, R.G., Smith, C.W., and Bhately, I.C., "Vortex-Lift Prediction for Complex Wing Planforms," *Journal of Aircraft*, Vol. 10, No. 6, June 1973, pp. 379–381.

12. Lamar, J.E., "Prediction of Vortex Flow Characteristics of Wings at Subsonic and Supersonic Speeds," *Journal of Aircraft*, Vol. 13, No. 7, July 1976.

13. Lamar, J.E., "Recent Studies of Subsonic Vortex Lift Including Parameters Affecting Stable Leading Edge Vortex Flow," *Journal of Aircraft*, Vol. 14, No. 12, December 1977.

14. Carlson, H.W. and Darden, C.M., "Attached Flow Numerical Methods for the Aerodynamics Design and Analysis of Vortex Flaps," NASA CP 2417, Paper 6.

15. Carlson, H.W. and Darden, C.M., "Applicability of Linearized-Theory Attached Flow Methods to Design and Analysis of Flap Systems at Low Speeds f or Thin Swept Wings With Sharp Leading Edges," NASA TP-2653, January 1987.

16. Carlson, H.W. and Walkly, K.B., "A Computer Program for Wing Subsonic Aerodynamic Performance Estimates Including Attainable Thrust and Vortex Lift Effects," NASA CR 3515.

17. Carlson, H.W. and Walkly, K.B., "An Aerodynamic Analysis Computer Program and Design Notes for Low-Speed Wing Flap Systems," NASA CR 3675, 1986.

18. Carlson, H.W., "The Design and Analysis of Simple Low-Speed Flap Systems with the Aid of Linearized Theory Computer Programs," NASA CR 3913, 1987.

19. Fox, C.H. Jr., "Prediction of Lift and Drag for Slender Sharp-Edge Delta Wings in Ground Proximity," NASA TN D-4891, January 1969.

20. Nuhait, A.O. and Mook, D.T., "Numerical Simulation of Wings in Steady and Unsteady Ground Effects," AIAA 88-2546-CP, 1988.

21. Peckham, D.H., "Low Speed Wind-Tunnel Tests on a Series of Uncambered Slender Pointed Wings With Sharp Edges," ARC R&M 3186, December 1958.

22. Coe, P.L., Jr. and Thomas, J.L., "Theoretical and Experimental Investigation of Ground-Induced Effects for a Low-Aspect-Ratio Highly Swept Arrow-Wing Configuration," NASA TP-1508, 1979.

23. McCormick, B.W., *Aerodynamics, Aeronautics, and Flight Mechanics*, John Wiley & Sons, N.Y., 1979.

24. Bertin, J.J. and Smith, M.L., *Aerodynamics for Engineers*, 2nd ed., Prentice-Hall, U.S., 1989, pp.257–295.

25. Bonner, E., Clever, W., and Dunn, K., *Aerodynamic Preliminary Analysis System II, Part I Theory*, North American Aircraft Operations, Rockwell International.

26. Davenport, E.E. and Huffman, J.K, "Experimental and Analytical Investigation of Subsonic Longitudinal and Lateral Aerodynamic Characteristics of Slender Sharp-Edge 74° Swept Wings," NASA TN D-6344, July 1971.

Appendix A: Computer Program Description

Programming Method

The computer program used to perform this investigation will be explained with continual reference to Figure A.1. Figure A.1 provides a general outline of the steps which the program follows when it is executed.

Before VLM-FIG is run, three separate programs must be defined and located in the same directory as VLM-FIG. The first of these, "blok.for" contains the variables in the common block as well as the variables which need to be dimensioned; it should not be altered. The second is "data.for." which contains all flap and wing geometrical data, the angle of attack, and the ground-effect information. This will need to be altered each time a different case is run. Table A.1 defines all of the variables needed for the most general cases to be run and Figure A.2 displays theses variables on a generalized wing. Note that an undeflected trailing-edge flap may be handled by setting DELTE = 0.0, by setting ITEFLEC = 0.0, or by setting NTEFL = 0. The same is true for a leading-edge flap if its analogous variables are similarly defined. The third program is "panel.for," which contains the wing and wake panelling information. Dense panelling results in excessive CPU time, especially if the ground-effect option is being used.

After subroutine VINITL has been called and the three ancillary programs have been included, the program calls subroutine GEOM. The main purpose of GEOM is to determine the x and y coordinates of the control points and of the

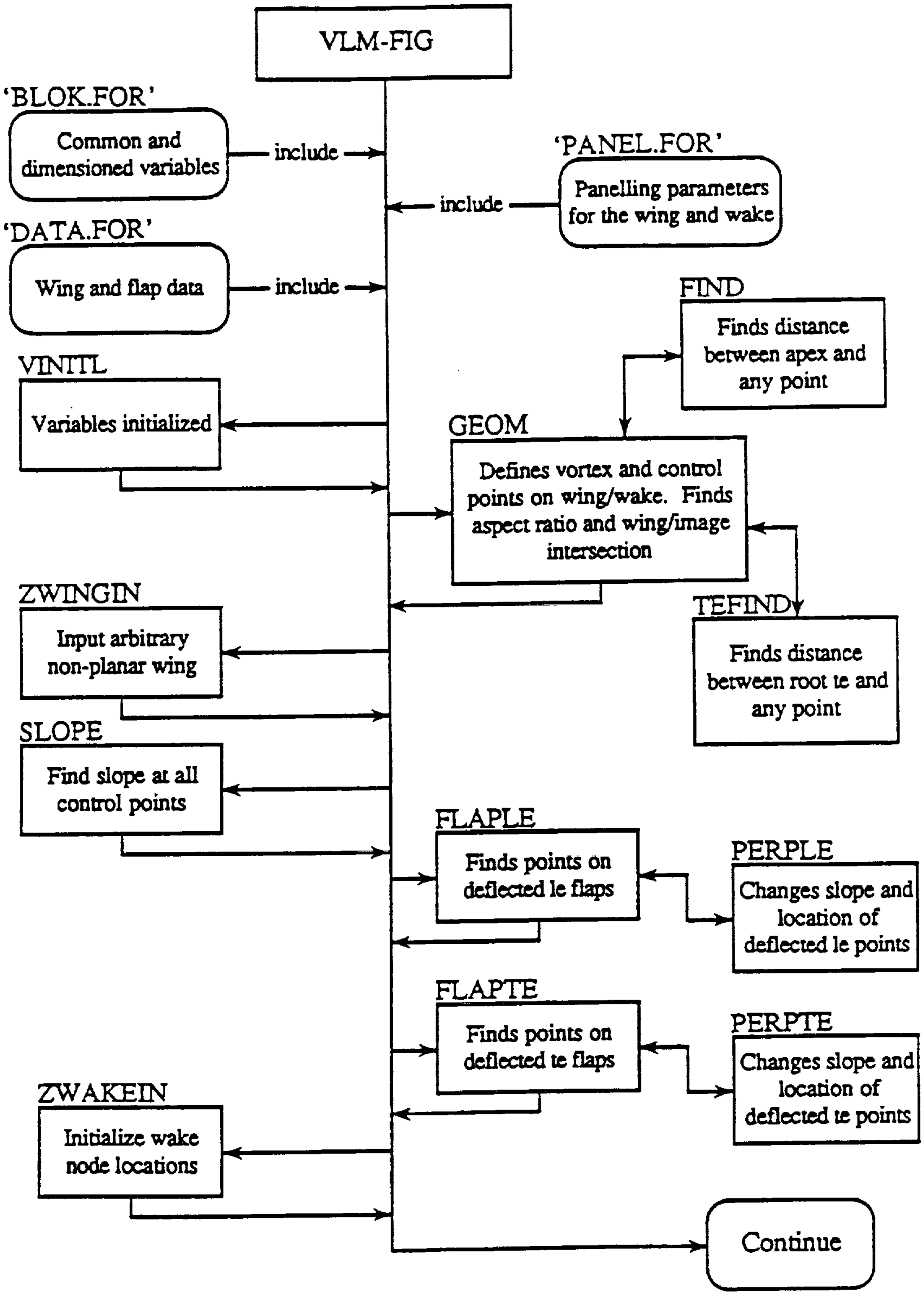
VLM-FIG
'BLOK.FOR'
Common and dimensioned variables
include
'PANEL.FOR'
Panelling parameters for the wing and wake
include
'DATA.FOR'
Wing and flap data
include
FIND
Finds distance between apex and any point
VINITL
Variables initialized
GEOM
Defines vortex and control points on wing/wake. Finds aspect ratio and wing/image intersection
ZWINGIN
Input arbitrary non-planar wing
TEFIND
Finds distance between root te and any point
SLOPE
Find slope at all control points
FLAPLE
Finds points on deflected le flaps
PERPLE
Changes slope and location of deflected le points
FLAPTE
Finds points on deflected te flaps
PERPTE
Changes slope and location of deflected te points
ZWAKEIN
Initialize wake node locations
Continue

Figure A.1. Flow Chart for VLM-FIG

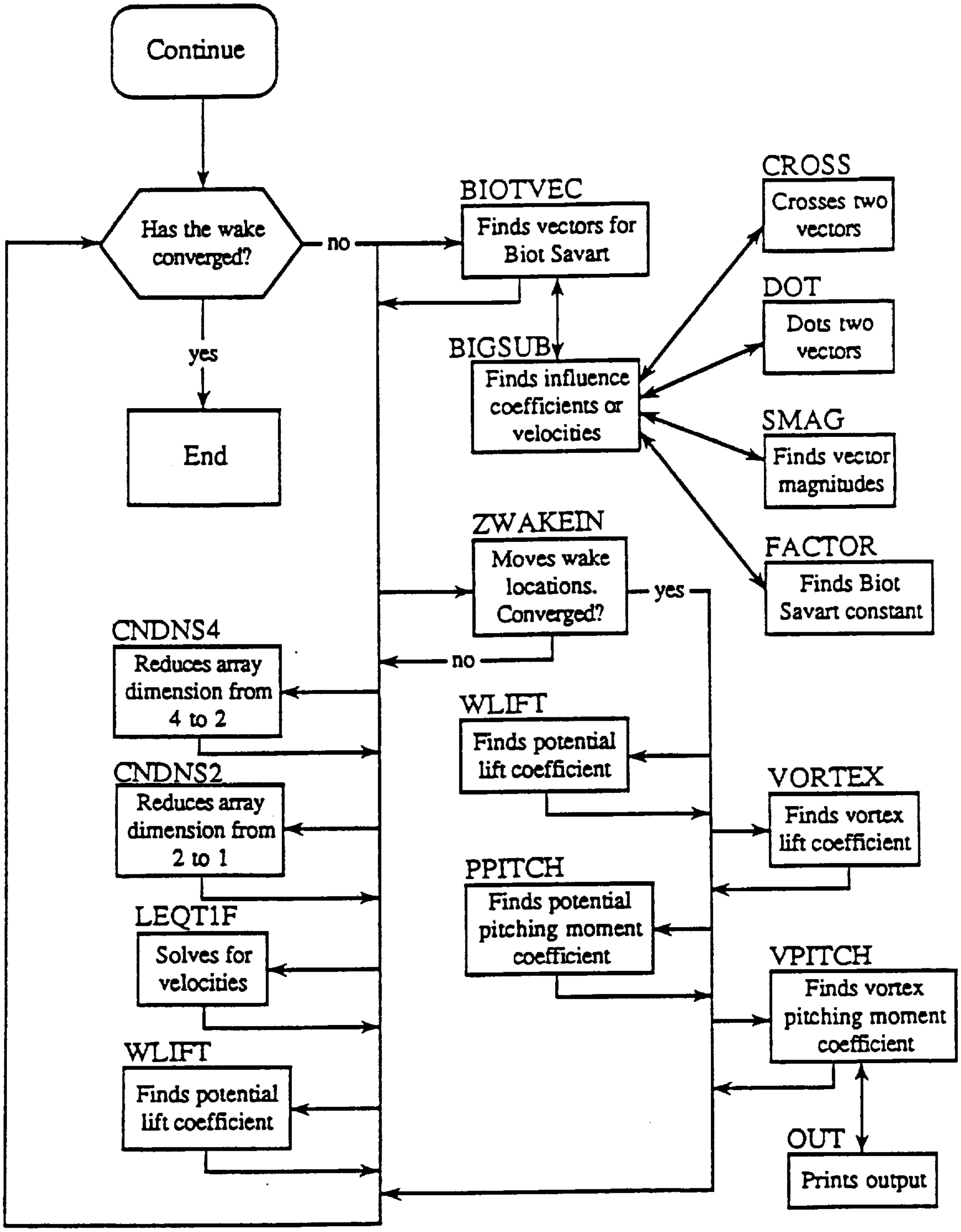
Continue
Has the wake converged?
no
yes
End
BIOTVEC
Finds vectors for Biot Savart
BIGSUB
Finds influence coefficients or velocities
CROSS
Crosses two vectors
DOT
Dots two vectors
SMAG
Finds vector magnitudes
FACTOR
Finds Biot Savart constant
ZWAKEIN
Moves wake locations. Converged?
yes
no
CNDNS4
Reduces array dimension from 4 to 2
CNDNS2
Reduces array dimension from 2 to 1
LEQT1F
Solves for velocities
WLIFT
Finds potential lift coefficient
WLIFT
Finds potential lift coefficient
VORTEX
Finds vortex lift coefficient
PPITCH
Finds potential pitching moment coefficient
VPITCH
Finds vortex pitching moment coefficient
OUT
Prints output

Figure A.1. Continued

Table A.1 A List of Variables to be Defined in 'data.for'

Leading Edge Variables	Description of Input
NLEFL	number of leading edge flaps
ILIGNL = 1	first inboard part of flap is aligned with the freestream
FANG(1)	hinge-line sweep angle of flap(1) (downward from horizontal)
FLASWE(1)	leading edge sweep angle of flap(1) (ccw from vertically down)
FLINX(1)	x coordinate of the most inboard hinge point
FLINY(1)	y coordinate of the most inboard hinge point
FOUTX(1)	x coordinate of the most outboard leading edge point
FOUTY(1)	y coordinate of the most outboard leading edge point
IDFLECT(1) = 1	flap(1) is deflected
DELLE(1)	flap(1) deflection in degrees
FANG(NLEFL)	hinge-line sweep angle of outermost flap
FLASWE(NLEFL)	leading edge sweep angle of outermost flap
FLINX(NLEFL)	x coordinate of the most inboard hinge point on outermost flap
FLINY(NLEFL)	y coordinate of the most inboard hinge point on outermost flap
FOUTX(NLEFL)	x coordinate of the most outboard le point on outermost flap
FOUTY(NLEFL)	y coordinate of the most outboard le point on outermost flap
IDFLECT(NLEFL) = 2	flap(NLEFL) is not deflected
DELLE(NLEFL) = 0.0	flap(NLEFL) is deflected 0.0 degrees
NUMEDGL = 2	numeric designation of the le flap on the wing edge

TrailingEdge Variables	Description of Input
NTEFL	number of trailing edge flaps
ILIGNT(1)	inboard part of first te flap is aligned with the freestream
TFANG(1)	hinge-line sweep angle of te flap(1) (downward from horizontal)
TFLASW(1)	te sweep angle of te flap(1) (downward from horizontal)
TFINX(1)	x coordinate of most inboard hinge point of TEF1
TFINY(1)	y coordinate of most inboard hinge point of TEF1
TOUTX(1)	x coordinate of most outboard point on te of TEF1
TOUTY(1)	y coordinate of most outboard point on te of TEF1
ITEFLEC(1)	TEFLAP(1) is deflected
DELTE(1)	TEFLAP(1) deflection in degrees (+ downward)
ILIGNT(NTEFL)	inboard part of outermost te flap is aligned with the freestream
TFANG(NTEFL)	hinge-line sweep angle of outermost te flap
TFLASW(NTEFL)	trailing edge sweep angle of outermost te flap
TFINX(NTEFL)	x coordinate of most inboard hinge point of outermost te flap
TFINY(NTEFL)	y coordinate of most inboard hinge point of outermost te flap
TOUTX(NTEFL)	x coordinate of most outboard te point of outermost te flap
TOUTY(NTEFL)	y coordinate of most outboard te point of outermost te flap
ITEFLEC(NTEFL)	outermost te flap is deflected
DELTE(NTEFL)	TEFLAP(NTEFL) deflection in degrees (+ downward)
NUMEDGT = 2	numeric designation of the te flap on the wing edge

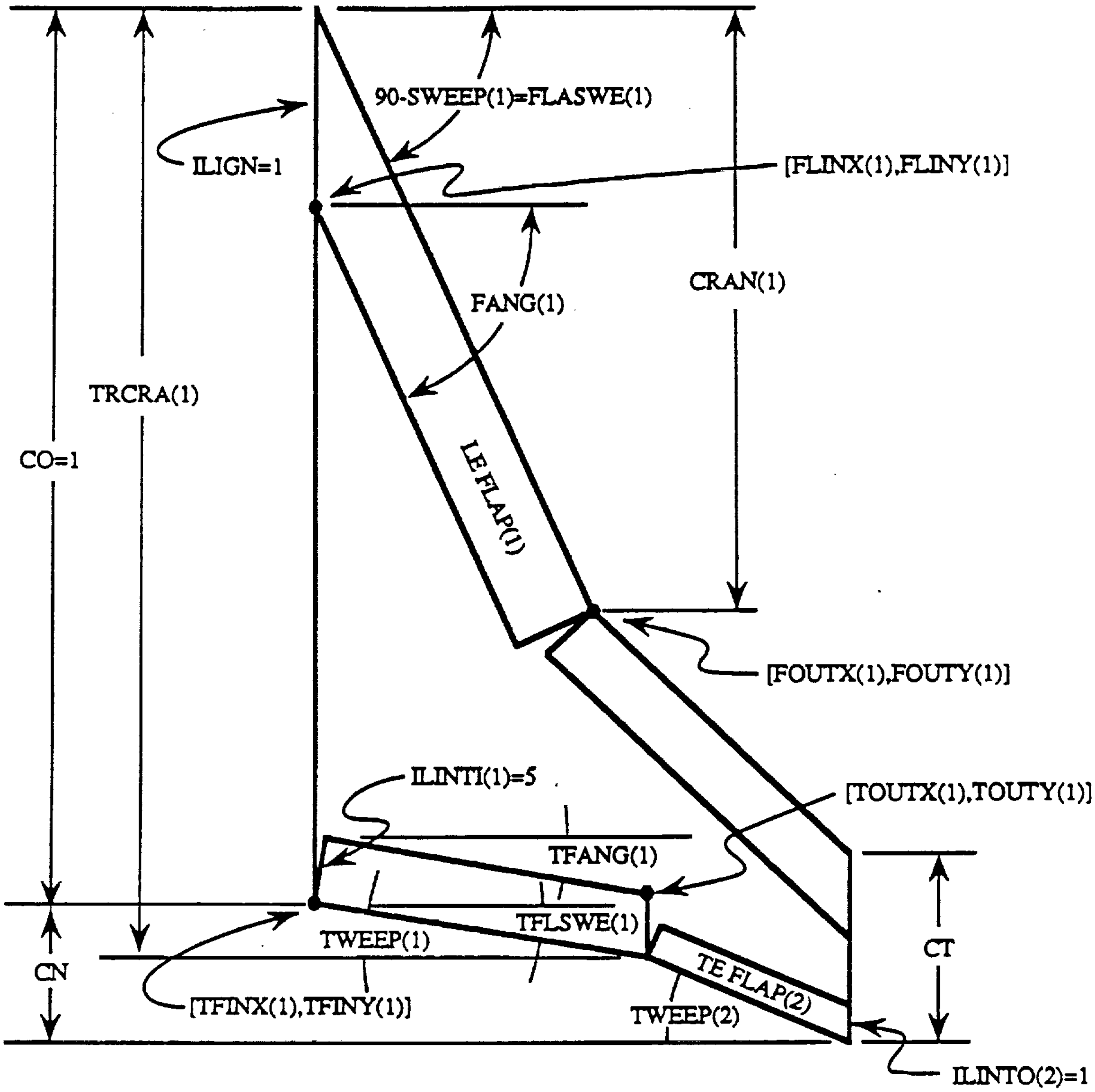

Figure A.2 Location of Input Parameters on a Generalized Wing

endpoint of the vortex segments which lie on the wing. The trailing-edge points where the vortex segments first touch the wake are also calculated. As a consequence of calculating these values, the aspect ratio, wing span, and chord at various spanwise locations are found.

Since the program allows for wings with up to nine cranks in the leading and trailing edges, it is necessary to determine the spanwise location of a point. The subroutines FIND and TEFIND determine the distance of the leading edge behind the apex and of the trailing edge behind the root chord. This enables the local chord to be calculated, which is critical to proper panelling.

If the ground effect option is being used, GEOM calculates the parameters which are used to locate the intersection of the root chords of the real wing and the image wing.

The subroutine ZWINGIN is intended to be used for manually entering the z coordinates of a nonplanar wing. By default, a planar wing is input. If the ZWINGIN option is chosen, the program will list the x and y coordinates of the control points and vortex segments. The user must then input the proper z coordinates. The subroutine SLOPE will set the slope of the wing to zero at each control point. For non-planar wings, the slope at each control point must be entered manually.

At this point, the program has determined the slope and location of each control point and has determined the locations of all wing vortex segments. If the wing is flapped, it must be determined which of these should be altered, and by how much.

Subroutine FLAPLE uses the points which were entered in "data.for" to define regions near the wing leading edge which are flapped, while subroutine FLAPTE uses the points to define regions near the wing trailing edge which are flapped. If a vortex segment endpoint lies on a flapped region, then it is connected to the hinge with a perpendicular line and this line is rotated about the hingeline until it reaches an angle which equals the flap deflection. Control points are moved similarly, and the slope of the wing is also changed by an amount equal to the flap deflection. Subroutines PERPLE and PERPTE are used for finding perpendicular distances and changing slopes of affected points.

The wing geometry is now fixed, but the wake geometry must be defined. Subroutine ZWAKEIN performs this function by defining each node in the wake as having the same *y*- and *z*-coordinates as the point on the trailing edge of the wing over which the segment's vortex line passed.

The next major portion of the program enables the wake to reach a relaxed position and it begins when the DO WHILE loop is first encountered.

There are two phases to the iteration scheme, the first occurs when the variable NEVEN equals 1 and the second occurs when the variable NEVEN equals 2. Phase 1 calculates influence coefficients, solves a set of simultaneous equations to find the circulation strengths, and then calculates the lift coefficient. Phase 2 uses the known values of circulation to calculate the downwash in the wake and then a subroutine is used to move the wake locations. Phase 1 is then re-entered so that the strengths of the circulations based on the new wake locations can be recalculated. The specific path which is followed to carry out this procedure is now described.

For Phase 1, subroutine BIOTVEC is called so that the endpoints of the vortex segments can be selected for analysis. Since the vorticies are horseshoe shaped, vortex segments will overlap near the rear portions of the wing which means that several vortex strengths share the same location on the wing and wake. Subroutine BIGSUB is then called so that the influence coefficient at each control point due to the particular segment may be found. These values are then added to the previous values of the influence coefficient of each control point. Since BIGSUB must calculate the effect for many segments ($\sim$ 1000) at all the control points, the only information about the segment which is saved is the segment bound vortex location and the location of the control point being influenced. This allows the versatility to standardize the calculations, particularly with regard to ground effect, but consequently only the summed influence coefficients, C_{mn} can be known after a segment has been processed by BIGSUB. Subroutine BIGSUB calls subroutines CROSS, SMAG, DOT, and FACTOR to cross two vectors, find the magnitude of a vector, dot two vectors, and calculate a factor needed for the Biot-Savart law, respectively. Subroutine ZWAKEIN is called next, but since the iteration is in Phase 1 control is returned to the main program and suroutines CNDNS4 and CNDNS2 are called.

CNDNS4 is used to condense a four dimensional array into a two dimensional array, while CNDNS2 is used to condense a two dimensional array into a one dimensional array. This is required since the canned subroutine LEQT1F solves a system of equations with one and two dimensional arrays. No information is gained or lost in this process, the control points and circulation strengths are simply identified using a different filing system.

Next, the boundary conditions are calculated, the linear equation solving subroutine LEQT1F is called, and WLIFT is called to calculate the lift coefficient. The iteration phase is changed from NEVEN = 1 to NEVEN = 2 and the second phase of the iteration is started.

Subroutine BIOTVEC begins the iteration phase by performing the same function as it did for Phase 1, namely, to send all of the vortex segments to subroutine BIGSUB for processing. Since the program is in Phase 2, BIGSUB calculates the influence coefficients as before, but since the circulation strengths are known from Phase 1, it proceeds to calculate the downwash at each node in the wake. Again, subroutines CROSS, SMAG, DOT, and FACTOR are called by BIGSUB.

ZWAKEIN is the next subroutine to be called, In this phase, it is the most versatile subroutine in the program. The first two roles of this subroutine are to define vectors from each node to the node just downstream of it and from each node to the new wake location of the node just downstream of it. Certain vector operations are performed on these vectors so that the angle between them can be calculated. The subroutine then checks if all of the nodes have angles which are within the termination limit.

If all angles are within this limit, a flag variable is set for termination and subroutines VORTEX, PPITCH, and VPITCH are called. These subroutines calculate the vortex lift, the potential pitching moment and the vortex pitching moment. Subroutine VPITCH then calls OUT which writes the output to a file, the details of this output will be explained in the next section.

If all angles are not within the termination limit, the flag variable is not set for termination, and control is soon returned to the main program where it will switch NEVEN and the program will re-enter Phase 1. Because of empirical evidence gathered during the program's development, there is an option which allows the program to be terminated even if all of the wake nodes have not converged. If more than ten full iterations, consisting of two phases each, have been performed, the program's flag variable is set to the termination value and the termination procedure described earlier is commenced. This option is most useful for high angles of attack and it does not alter the results.

Output

Subroutine OUT is used to generate the output of VLM-FIG. The angle of attack, the taper ratio, the notch ratio and the leading-edge sweep angles are the parameters which are echoed as a check that they were properly entered in the program. Rather than echo all of the other input parameters, the aspect ratio is printed. The aspect ratio is dependent on sweep, crank location, notches, and cropped tips and therefore, reveals a good deal of information about the program. If the aspect ratio is too low, for example, it may indicate that the tip chord which was entered was too large.

The potential, vortex, and total lift coefficients are also tabulated. In addition, the potential, vortex, and total lift coefficients as well as the vortex and potential constants are calculated. The total lift and moment coefficients are the two values which are the most useful since they represent the aerodynamic characteristics which would be measured in the wind tunnel or in flight. The other values are most useful for comparing the predictions of VLM-FIG with the predictions of other programs.

Appendix B: Computer Program Listing

```
Filename: TOO.FOR
C
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
C 28 IN
INCLUDE 'PANEL.FOR'
C 4 IN
CALL VINITL
INCLUDE 'DATA.FOR'
CALL GEOM
CALL ZWINGIN
CALL SLOPE
CALL FLAPLE
CALL FLAPTE
CALL ZWAKEIN
DO WHILE (ITER .NE. 0)
NCOUNT=NCOUNT+1
WRITE(6,*) 'NCOUNT',NCOUNT
CALL BIOTVEC
CALL ZWAKEIN
IF (NEVEN .EQ. 1) THEN
CALL CNDNS4
CALL CNDNS2
DO 100 I=1,KAU
B(I)=UINF*DSIN(ALPHA-DELTS(I))*DCOS(DI)
100 CONTINUE
CALL LEQT1F(A,M,N,IA,B,IDGT,WKAREA,IER)
CALL WLIFT
END IF
IF (NEVEN .EQ. 1) THEN
NEVEN=2
ELSE
NEVEN=1
END IF
END DO
STOP
END
C
C SUBROUTINES SUBROUTINES SUBROUTINES
C
SUBROUTINE VINITL
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
C
C INITIALIZE THE VARIABLES
C
PI=DACOS(-1.0D0)
UINF=1.0
NCHORD1=NCHORD+1
```

```
NSPAN1=NSPAN+1
NWAKE1=NWAKE+1
LINES=NCHORD+NWAKE1
KAU=NCHORD*NSPAN
KAOI=NSPAN*NWAKE1
ITER=1
IER=0
IDGT=0
NCOUNT=0
NEVEN=1
M=1
N=KAU
IA=100
RETURN
END
C
C
C
SUBROUTINE GEOM
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
C
LEEDM1=0
IF (LEEDS .GE. 2) THEN
LEEDM1=LEEDS-1
DO 500 I=1,LEEDM1
I1=I+1
ADAR(I)=(1./DTAN(SWEEP(I))-1./DTAN(SWEEP(I1)))*CRAN(I)**2
CO(I1)=CO(I)-(1./DTAN(SWEEP(I))-1./DTAN(SWEEP(I1)))*CRAN(I)
500 CONTINUE
END IF
BTRI=2.0*(CO(LEEDS)+CN)*DTAN(SWEEP(LEEDS))
YCUT=CT*DTAN(SWEEP(LEEDS))
BCROP=(BTRI/2.0-YCUT)*2.0
AIRCUT=.5*CT*YCUT
AIRNOT=.5*CN*(BCROP/2.0)
AIRTRI=.5*(CO(LEEDS)+CN)*(BTRI/2.0)
IF (LEEDS .EQ. 1) THEN
CRAN(1)=BCROP/2.0
END IF
IF (NTREDS .EQ. 1) THEN
TRCRA(1)=BCROP/2.0
TRSPN(1)=BCROP/2.0
END IF
HOVB=HOVB*BCROP
SECDIS=HOVB/DSIN(ALPHA)
XSEC=1.0+SECDIS
NTREM1=0.0
IF (NTREDS .GE. 2) THEN
AIRNOT=2.0*AIRNOT
NTREM1=NTREDS-1
```

```
DO 600 I=1,NTREM1
I1=I+1
IM1=I-1
C
C CONVERT CHORDWISE TRCRA(I) TO SPANWISE TRSPN(I)
C WHY NOT JUST PUT IT IN AS A SPANWISE LOCATION?
C
IF (I .EQ. 1) THEN
TRTR=TRCRA(I)-CO(1)
IF (TWEEP(I) .LT. .001) THEN
TRSPN(I)=SPOCK
ELSE
TRSPN(I)=TRTR/DTAN(TWEEP(I))
END IF
ELSE
TRTR=TRCRA(I)-TRCRA(IM1)
TRSPN(I)=TRSPN(IM1)+TRTR/DTAN(TWEEP(I))
END IF
IF (TWEEP(I) .GT. .001) THEN
TRAR=(.5/DTAN(TWEEP(I))-.5/DTAN(TWEEP(I1)))*TRTR**2
END IF
AIRNOT=AIRNOT-TRAR
600 CONTINUE
TETRI=.5*CN*CN/DTAN(TWEEP(NTREDS))
AIRNOT=AIRNOT-TETRI
END IF
AIREA=2.0*(AIRTRI-AIRNOT-AIRCUT)
C
C
WRITE(6,*)'AREA OF WING',AIREA
IF (LEEDS .GE. 2) THEN
DO 700 I=1,LEEDM1
AIREA=AIREA+ADAR(I)
700 CONTINUE
END IF
AR=BCROP*BCROP/AIREA
WRITE(6,*)'ASPECT RATIO',AR
DELY=BCROP/(2.*FLOAT(NSPAN))
WEIGHT=4.0*DELY/AIREA
C
C CALCULATE THE VORTEX SEGMENT AND CONTROL POINT X,Y LOCATIONS
C
DO 910 I=1,NSPAN
DO 900 K=1,NCHORD
IM1=I-1
KM1=K-1
IF (I .EQ. 1 ) THEN
YCP(K,I)=DELY/2.0
YVLLFT(K,I)=YCP(K,I)-DELY/2.0
YVLRGT(K,I)=YCP(K,I)+DELY/2.0
ELSE
```

```
YCP(K,I)=YCP(K,IM1)+DELY
YVLLFT(K,I)=YCP(K,I)-DELY/2.0
YVLRGT(K,I)=YCP(K,I)+DELY/2.0
END IF
C
C CALCULATE THE CHORD AT A GIVEN Y LOCATION
C
DO 800 ICE=1,3
IF (ICE .EQ. 1) THEN
CORE=YCP(K,I)
CALL FOUND
XLOST=ALOSS
CALL TEFIND
XEXTRA=EXTRA
END IF
IF (ICE .EQ. 2) THEN
CORE=YCP(K,I)-DELY/2.0
CALL FOUND
XLOSTL=ALOSS
CALL TEFIND
XEXTRAL=EXTRA
END IF
IF (ICE .EQ. 3) THEN
CORE=YCP(K,I)+DELY/2.0
CALL FOUND
XLOSTR=ALOSS
CALL TEFIND
XEXTRAR=EXTRA
END IF
800 CONTINUE
CRDATY(I)=1.0+XEXTRA-XLOST
CRDATL=1.0+XEXTRAL-XLOSTL
CRDATR=1.0+XEXTRAR-XLOSTR
C
C CALCULATE DELTA X AT LEFT,RIGHT AND AT CONTROL POINT
C
DELX=CRDATY(I)/(FLOAT(NCHORD))
DELXL=CRDATL/(FLOAT(NCHORD))
DELXR=CRDATR/(FLOAT(NCHORD))
IF (K .EQ. 1) THEN
XCP(K,I)=.75*DELX+(XLOST)
XVLLFT(K,I)=.25*DELXL+(XLOSTL)
XVLRGT(K,I)=.25*DELXR+(XLOSTR)
ELSE
XCP(K,I)=XCP(KM1,I)+DELX
XVLLFT(K,I)=XVLLFT(KM1,I)+DELXL
XVLRGT(K,I)=XVLRGT(KM1,I)+DELXR
END IF
YTEL(I)=YVLLFT(K,I)
YTER(I)=YVLRGT(K,I)
XTEL(I)=CRDATL+XLOSTL
```

```
XTER(I)=CRDATR+XLOSTR
900 CONTINUE
910 CONTINUE
RETURN
END
C
C
C
SUBROUTINE FOUND
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
IF (CORE .LE. CRAN(1) .OR. (CRAN(1)-CORE) .LT. .0000001) THEN
ALOSS=CORE/DTAN(SWEEP(1))
XWEE=SWEEP(1)
ELSE
FIND=1.0
MOT=1
DO WHILE (FIND .LT. 10.0)
MOT=MOT+1
MOTM1=MOT-1
IF (MOT .LE. LEEDM1) THEN
YKRANK=CRAN(MOT)
ELSE
YKRANK=BCROP
END IF
IF (CORE .LE. YKRANK) THEN
IF (FCOR .GT. CRAN(MOTM1)) THEN
FIND=25.0
XWEE=SWEEP(MOT)
DO 1000 MOE=1,MOTM1
MOEM1=MOE-1
IF (MOE .EQ. 1) THEN
ALOSS=CRAN(1)/DTAN(SWEEP(1))
ELSE
ALOSS=ALOSS+(CRAN(MOE)-CRAN(MOEM1))/DTAN(SWEEP(MOE))
END IF
1000 CONTINUE
ALOSS=ALOSS+(CORE-CRAN(MOTM1))/DTAN(SWEEP(MOT))
END IF
END IF
END DO
END IF
RETURN
END
C
C
C
SUBROUTINE TEFIND
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
IF (CORE .LE. TRSPN(1)) THEN
```

```
EXTRA=CORE*DTAN(TWEEP(1))
ELSE
FIND=1.0
MOT=1
DO WHILE (FIND .LT. 10.0)
MOT=MOT+1
MOTM1=MOT-1
IF (MOT .LE. NTREM1) THEN
YKRANK=TRSPN(MOT)
ELSE
YKRANK=BCROP
END IF
IF (CORE .LE. YKRANK) THEN
IF (CORE .GT. TRSPN(MOTM1)) THEN
FIND=25.0
DO 1100 MOE=1,MOTM1
MOEM1=MOE-1
IF (MOE .EQ. 1) THEN
EXTRA=TRSPN(1)*DTAN(TWEEP(1))
ELSE
EXTRA=EXTRA+(TRSPN(MOE)-TRSPN(MOEM1))*DTAN(TWEEP(MOE))
END IF
1100 CONTINUE
EXTRA=EXTRA+(CORE-TRSPN(MOTM1))*DTAN(TWEEP(MOT))
END IF
END IF
END DO
END IF
RETURN
END
C
C
C
SUBROUTINE ZWINGIN
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
IF (NEVEN .EQ. 10) THEN
C WRITE(6,*)'XTEL ','YTEL ','FIND Z'
DO 4400 I=1,NSPAN
C WRITE(6*) XTEL(I),YTEL(I)
4400 CONTINUE
C WRITE(6,*),'XTER ','YTER ','FIND Z'
DO 4410 I=1,NSPAN
C WRITE(6,*),XTER(I),YTER(I)
4410 CONTINUE
DO 4420 I=1,NSPAN
C READ(6,*)ZTEL(I)
ZTEL(I)=0.0
4420 CONTINUE
DO 4430 I=1,NSPAN
C READ(6,*)ZTER(I)
```

```
      ZTER(I)=0.0
      4430 CONTINUE
      C
      C NOW THE Z COORDINATES OF THE LEFT AND THEN RIGHT VORTEX SEGMENTS
WILL
      C BE CALCULATED
      C
      C WRITE(6,*)'XVLLFT','YVLLFT','FIND Z'
      DO 4440 I=1,NSPAN
      DO 4440 K=1,NCHORD
      C WRITE(6,*)XVLLFT(K,I),YVLLFT(K,I)
      4440 CONTINUE
      C WRITE(6,*)'XCP ','YCP ','FIND Z'
      DO 4450 I=1,NSPAN
      DO 4450 K=1,NCHORD
      C WRITE(6,*)XCP(K,I),YCP(K,I)
      4450 CONTINUE
      C WRITE(6,*)'XVLRGT','YVLRGT','FIND Z'
      DO 4460 I=1,NSPAN
      DO 4460 K=1,NCHORD
      C WRITE(6,*),XVLRGT(K,I),YVLRGT(K,I)
      4460 CONTINUE
      DO 4470 I=1,NSPAN
      DO 4470 K=1,NCHORD
      C READ(6,*)ZVLLFT(K,I)
      ZVLLFT(K,I)=0.0
      4470 CONTINUE
      DO 4480 I=1,NSPAN
      DO 4480 K=1,NCHORD
      C READ(6,*)ZVLRGT(K,I)
      ZVLRGT(K,I)=0.0
      4480 CONTINUE
      DO 4490 I=1,NSPAN
      DO 4490 K=1,NCHORD
      C READ(6,*)ZCP(K,I)
      ZCP(K,I)=0.0
      4490 CONTINUE
      END IF
      RETURN
      END
      C
      C
      C
      SUBROUTINE SLOPE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INCLUDE 'BLOK.FOR'
      DO 50 I=1,NSPAN
      DO 50 K=1,NCHORD
      DZDX(K,I)=0.0
      50 CONTINUE
      RETURN
```

```
END
C
C
C
SUBROUTINE FLAPLE
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
LECT=0
DO 2000 JA=1,NLEFL
IF (IDFLECT(JA) .EQ. 1) THEN
LECT=LECT+1
NUML(LECT)=JA
WRITE(6,*)'NUML(LECT),LECT',NUML(LECT),LECT
END IF
2000 CONTINUE
IF (LECT .NE. 0) THEN
DO 2200 JI=1,LECT
JA=NUML(JI)
JA1=JA+1
FUNPTX=(FOUTY(JA)-FLINY(JA))*DTAN(FANG(JA))+FLINX(JA)
FUNPTX=FUNPTX+FOUTX(JA)*DTAN(FANG(JA))*DTAN(FANG(JA))
FUNPTX=FUNPTX/(1.0+DTAN(FANG(JA))*DTAN(FANG(JA)))
FUNPTY=FLINY(JA)+(FUNPTX-FLINX(JA))/DTAN(FANG(JA))
C
FLUX=FLINY(JA)-FOUTY(JA)+FLINX(JA)*DTAN(FANG(JA))
FLUX=FLUX+FOUTX(JA)*DTAN(FLASWE(JA))
FLUX=FLUX/(DTAN(FANG(JA))+DTAN(FLASWE(JA)))
FLUY=FLINY(JA)+(FLINX(JA)-FLUX)*DTAN(FANG(JA))
DO 2100 I=1,NSPAN
DO 2100 K=1,NCHORD
DO 2100 LOW=1,3
IF (LOW .EQ. 1) THEN
XPOIN=XCP(K,I)
YPOIN=YCP(K,I)
ZPOIN=ZCP(K,I)
END IF
IF (LOW .EQ. 2) THEN
XPOIN=XVLLFT(K,I)
YPOIN=YVLLFT(K,I)
ZPOIN=ZVLLFT(K,I)
END IF
IF (LOW .EQ. 3) THEN
XPOIN=XVLRGT(K,I)
YPOIN=YVLRGT(K,I)
ZPOIN=ZVLRGT(K,I)
END IF
C
C REGION 1
C
IF (ILIGN .EQ. 1) THEN
IF (JA .EQ. 1) THEN
IF (XPOIN .LE. FLINX(JA)) THEN
CALL PERPLE
END IF
END IF
END IF
IF (JA .NE. 1 .OR. ILIGN .NE. 1) THEN
IF (XPOIN .LE. FLINX(JA)) THEN
IF (XPOIN .GE. FLUX) THEN
YUM=FLINY(JA)+(FLINX(JA)-XPOIN)*DTAN(FANG(JA))
IF (YPOIN .GE. YUM) THEN
CALL PERPLE
END IF
END IF
END IF
END IF
C
C REGION 2
C
IF (XPOIN .GE. FLINX(JA)) THEN
IF (XPOIN .LE. FOUTX(JA)) THEN
YUM=FLINY(JA)+(XPOIN-FLINX(JA))/DTAN(FANG(JA))
IF (YPOIN .GE. YUM) THEN
CALL PERPLE
END IF
END IF
END IF
C
C REGION 3
C
IF (XPOIN .GE. FOUTX(JA)) THEN
IF (NUMEDGL .EQ. NUML(JI)) THEN
XUM=FLINX(JA)+(FOUTY(JA)-FLINY(JA))*DTAN(FANG(JA))
IF (XPOIN .LE. XUM) THEN
YUM=FOUTY(JA)+(XPOIN-XUM)/DTAN(FANG(JA))
IF (YPOIN .GE. YUM) THEN
CALL PERPLE
END IF
END IF
ELSE
IF (XPOIN .LE. FUNPTX) THEN
TEMX=FLUX
TEMY=FLUY
FLUX=FLINY(JA1)-FOUTY(JA1)+FLINX(JA1)*DTAN(FANG(JA1))
FLUX=FLUX+FOUTX(JA1)*DTAN(FLASWE(JA1))
FLUX=FLUX/(DTAN(FANG(JA1))+DTAN(FLASWE(JA1)))
FLUY=FLINY(JA1)+(FLINX(JA1)-FLUX)*DTAN(FANG(JA1))
YUM=(XPOIN-FLINX(JA))/DTAN(FANG(JA))+FLINY(JA)
IF (YPOIN .GT. YUM) THEN
IF (XPOIN .LE. FLUX) THEN
CALL PERPLE
ELSE
```

```
YUM=FLUY-(XPOIN-FLUX)*DTAN(FANG(JA1))
IF (YPOIN .LE. YUM) THEN
CALL PERPLE
END IF
END IF
END IF
FLUX=TEMX
FLUY=TEMY
END IF
END IF
END IF
IF (LOW .EQ. 1) THEN
ZCP(K,I)=ZPOIN
END IF
IF (LOW .EQ. 2) THEN
ZVLLFT(K,I)=ZPOIN
END IF
IF (LOW .EQ. 3) THEN
ZVLRGT(K,I)=ZPOIN
END IF
2100 CONTINUE
2200 CONTINUE
END IF
RETURN
END
C
C
C
SUBROUTINE PERPLE
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
C
PERX=(YPOIN-FUNPTY)*DTAN(FANG(JA))+FUNPTX
PERX=PERX+XPOIN*DTAN(FANG(JA))*DTAN(FANG(JA))
PERX=PERX/(1.0+DTAN(FANG(JA))*DTAN(FANG(JA)))
PERY=YPOIN+(XPOIN-PERX)*DTAN(FANG(JA))
DISTN=((XPOIN-PERX)**2+(YPOIN-PERY)**2)**.5
ZPOIN=ZPOIN-DISTN*DCOS(DI)*DSIN(DELLE(JA))
IF (LOW .EQ. 1) THEN
DZDX(K,I)=DZDX(K,I)+DTAN(DELLE(JA))*DCOS(FANG(JA))
END IF
RETURN
END
C
C
C
SUBROUTINE FLAPTE
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
ITECT=0
DO 3000 JA=1,NTEFL
```

```
IF (ITEFLEC(JA) .EQ. 1) THEN
ITECT=ITECT+1
NUMT(ITECT)=JA
END IF
WRITE(6,*)'NUMT(ITECT),ITECT',NUMT(ITECT),ITECT
3000 CONTINUE
IF (ITECT .NE. 0) THEN
DO 3200 JI=1,ITECT
JA=NUMT(JI)
JA1=JA+1
C
IF (ILINTI(JA) .EQ. 1) THEN
TIX=(TFINY(JA)-TOUTY(JA))*DTAN(TFANG(JA))+TOUTX(JA)
TIY=TFINY(JA)
ELSE
TIX=TOUTX(JA)+(TFINY(JA)-TOUTY(JA))*DTAN(TFANG(JA))
TIX=TIX+TFINX(JA)*DTAN(TFANG(JA))*DTAN(TFANG(JA))
TIX=TIX/(1.0+DTAN(TFANG(JA))*DTAN(TFANG(JA)))
TIY=TFINY(JA)+(TFINX(JA)-TIX)*DTAN(TFANG(JA))
END IF
IF (ILINTO(JA) .EQ. 1) THEN
TOY=TOUTY(JA)
TOX=TOUTX(JA)+(TOUTY(JA)-TFINY(JA))*DTAN(TFLSWE(JA))
ELSE
TOX=(TOUTY(JA)-TFINY(JA))*DTAN(TFLSWE(JA))+TFINX(JA)
TOX=TOX+TOUTX(JA)*DTAN(TFANG(JA))*DTAN(TFLSWE(JA))
TOX=TOX/(1.0+DTAN(TFANG(JA))*DTAN(TFLSWE(JA)))
TOY=TOUTY(JA)+(TOUTX(JA)-TOX)*DTAN(TFANG(JA))
END IF
C
DO 3160 I=1,NSPAN
DO 3140 K=1,NCHORD
DO 3120 LOT=1,5
IF (LOT .EQ. 1) THEN
XPOIN=XCP(K,I)
YPOIN=YCP(K,I)
ZPOIN=ZCP(K,I)
END IF
IF (LOT .EQ. 2) THEN
XPOIN=XVLLFT(K,I)
YPOIN=YVLLFT(K,I)
ZPOIN=ZVLLFT(K,I)
END IF
IF (LOT .EQ. 3) THEN
XPOIN=XVLRGT(K,I)
YPOIN=YVLRGT(K,I)
ZPOIN=ZVLRGT(K,I)
END IF
IF (LOT .EQ. 4) THEN
IF (K .EQ. 5) THEN
XPOIN=XTEL(I)
```

```
YPOIN=YTEL(I)
ZPOIN=ZTEL(I)
END IF
END IF
IF (LOT .EQ. 5) THEN
IF (K .EQ. 5) THEN
XPOIN=XTER(I)
YPOIN=YTER(I)
ZPOIN=ZTER(I)
END IF
END IF
C
XUM=TIX+(YPOIN-TIY)*DTAN(TFANG(JA))
IF (XPOIN .GE. XUM) THEN
IF (ILINTI(JA) .EQ. 1) THEN
IF (YPOIN .GE. TIY) THEN
IF (ILINTO(JA) .EQ. 1) THEN
IF (YPOIN .LE. TOY) THEN
CALL PERPTE
NTMOVE=1
END IF
ELSE
YUM=TOY+(TOX-XPOIN)*DTAN(TFANG(JA))
IF (YPOIN .LE. YUM) THEN
CALL PERPTE
NTMOVE=1
END IF
END IF
END IF
ELSE
YUM=TFINY(JA)+(TFINX(JA)-XPOIN)*DTAN(TFANG(JA))
IF (YPOIN .GE. YUM) THEN
IF (ILINTO(JA) .EQ. 1) THEN
IF (YPOIN .LE. TOY) THEN
CALL PERPTE
NTMOVE=1
END IF
ELSE
YUM=TOY+(TOX-XPOIN)*DTAN(TFANG(JA))
IF (YPOIN .LE. YUM) THEN
CALL PERPTE
NTMOVE=1
END IF
END IF
END IF
END IF
END IF
C
IF (LOT .EQ. 1) THEN
ZCP(K,I)=ZPOIN
END IF
```

```
IF (LOT .EQ. 2) THEN
ZVLLFT(K,I)=ZPOIN
END IF
IF (LOT .EQ. 3) THEN
ZVLRGT(K,I)=ZPOIN
END IF
IF (LOT .EQ. 4) THEN
IF (K .EQ. 5) THEN
ZTEL(I)=ZPOIN
END IF
END IF
IF (LOT .EQ. 5) THEN
IF (K .EQ. 5) THEN
ZTER(I)=ZPOIN
END IF
END IF
3120 CONTINUE
3140 CONTINUE
3160 CONTINUE
3200 CONTINUE
END IF
RETURN
END
C
C
C
SUBROUTINE PERPTE
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
C
PERX=TOUTX(JA)+(YPOIN-TOUTY(JA))*DTAN(TFANG(JA))
PERX=PERX+XPOIN*DTAN(TFANG(JA))*DTAN(TFANG(JA))
PERX=PERX/(1.0+DTAN(TFANG(JA))*DTAN(TFANG(JA)))
PERY=YPOIN+(XPOIN-PERX)*DTAN(TFANG(JA))
DISTN=((XPOIN-PERX)**2+(YPOIN-PERY)**2)**.5
ZPOIN=ZPOIN-DISTN*DCOS(DI)*DSIN(DELTE(JA))
IF (LOT .EQ. 1) THEN
DZDX(K,I)=DZDX(K,I)-DTAN(DELTE(JA))*DCOS(TFANG(JA))
END IF
RETURN
END
C
C
C
SUBROUTINE ZWAKEIN
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
DIMENSION VEC1L(3,20),VEC1R(3,20),VEC2L(3,20),VEC2R(3,20),
1 TEANGL(20),TEANGR(20),VL1MAG(20),VR1MAG(20),VL2MAG(20),
2 VR2MAG(20),XOWL(20,10),YOWL(20,10),ZOWL(20,10),XOWR(20,10),
3 YOWR(20,10),ZOWR(20,10),NODER(20,10),NODEL(20,10)
```

```
C
G=1.0
NSUM=NWAKE1*NSPAN
COD=(PI/180.0)*.01
IF (NCOUNT .EQ. 0) THEN
DO 4550 I=1,NSPAN
DO 4550 K=1,NWAKE1
C
C WAKE VORTEX LINE LOCATIONS
C
XWVLLF(K,I)=XTEL(I)+WAKDEL*FLOAT(K)
XOWL(K,I)=XWVLLF(K,I)
YWVLLF(K,I)=YTEL(I)
YOWL(K,I)=YWVLLF(K,I)
ZWVLLF(K,I)=ZTEL(I)
ZOWL(K,I)=ZWVLLF(K,I)
XWVLRT(K,I)=XTER(I)+WAKDEL*FLOAT(K)
XOWR(K,I)=XWVLRT(K,I)
YWVLRT(K,I)=YTER(I)
YOWR(K,I)=YWVLRT(K,I)
ZWVLRT(K,I)=ZTER(I)
ZOWR(K,I)=ZWVLRT(K,I)
4550 CONTINUE
END IF
C
IF (NEVEN .EQ. 2) THEN
C
C DETERMINE VECTOR 1 ON THE LEFT AND RIGHT OF THE PANEL
C
DO 4580 I=1,NSPAN
DO 4570 K=1,NWAKE1
I1=I+1
IM1=I-1
KM1=K-1
KK=K
II=I
IF (K .EQ. 1) THEN
VEC1L(1,I)=XWVLLF(K,I)-XTEL(I)
VEC1L(2,I)=YWVLLF(K,I)-YTEL(I)
VEC1L(3,I)=ZWVLLF(K,I)-ZTEL(I)
VEC1R(1,I)=XWVLRT(K,I)-XTER(I)
VEC1R(2,I)=YWVLRT(K,I)-YTER(I)
VEC1R(3,I)=ZWVLRT(K,I)-ZTER(I)
ELSE
VEC1L(1,I)=XWVLLF(K,I)-XOWL(KM1,I)
VEC1L(2,I)=YWVLLF(K,I)-YOWL(KM1,I)
VEC1L(3,I)=ZWVLLF(K,I)-ZOWL(KM1,I)
VEC1R(1,I)=XWVLRT(K,I)-XOWR(KM1,I)
VEC1R(2,I)=YWVLRT(K,I)-YOWR(KM1,I)
VEC1R(3,I)=ZWVLRT(K,I)-ZOWR(KM1,I)
END IF
```

```
C
C DETERMINE THE VECTORS TO THE NEW LOCATIONS
C
VEC2L(1,I)=DCOS(ALPHA)+COSUMX(K,I)
VEC2L(2,I)=COSUMY(K,I)
VEC2L(3,I)=DSIN(ALPHA)+COSUMZ(K,I)
VEC2R(1,I)=DCOS(ALPHA)+COSUMX(K,I1)
VEC2R(2,I)=COSUMY(K,I1)
VEC2R(3,I)=DSIN(ALPHA)+COSUMZ(K,I1)
C
C
C FIND VECTOR MAGNITUDES
C
VL1MAG(I)=(VEC1L(1,I)**2+VEC1L(2,I)**2+
1 VEC1L(3,I)**2)**.5
VR1MAG(I)=(VEC1R(1,I)**2+VEC1R(2,I)**2+
1 VEC1R(3,I)**2)**.5
VL2MAG(I)=(VEC2L(1,I)**2+VEC2L(2,I)**2+
1 VEC2L(3,I)**2)**.5
VR2MAG(I)=(VEC2R(1,I)**2+VEC2R(2,I)**2+
1 VEC2R(3,I)**2)**.5
C
C MAKE VECTOR 2 A UNIT VECTOR
C
VEC2L(1,I)=VEC2L(1,I)/VL2MAG(I)
VEC2L(2,I)=VEC2L(2,I)/VL2MAG(I)
VEC2L(3,I)=VEC2L(3,I)/VL2MAG(I)
VEC2R(1,I)=VEC2R(1,I)/VR2MAG(I)
VEC2R(2,I)=VEC2R(2,I)/VR2MAG(I)
VEC2R(3,I)=VEC2R(3,I)/VR2MAG(I)
C
C MOVE WAKE LOCATIONS, STRETCH OR SHRINK VECTOR COMPONENTS
C
XWVLLF(K,I)=XOWL(K,I)
YWVLLF(K,I)=YOWL(K,I)+VEC2L(2,I)*WAKDEL/VEC2L(1,I)
ZWVLLF(K,I)=ZOWL(K,I)+VEC2L(3,I)*WAKDEL/VEC2L(1,I)
XWVLRT(K,I)=XOWR(K,I)
YWVLRT(K,I)=YOWR(K,I)+VEC2R(2,I)*WAKDEL/VEC2R(1,I)
ZWVLRT(K,I)=ZOWR(K,I)+VEC2R(3,I)*WAKDEL/VEC2R(1,I)
C
C USE DOT PRODUCT TO FIND ANGLES BETWEEN OLD AND NEW WAKE NODES
C
CHECKL=(VEC1L(1,I)*VEC2L(1,I)+VEC1L(2,I)*
1 VEC2L(2,I)+VEC1L(3,I)*VEC2L(3,I))/(VL1MAG(I))
CHEK=DABS(1.0-CHECKL)
IF (CHEK .LT. 0.000000001) THEN
TEANGL(I)=0.0
ELSE
TEANGL(I)=DACOS(CHECKL)
END IF
C WRITE(9,*)'TE ANGLE V1-V2 ON LEFT',TEANGL(I)*180./PI
```

```
C
CHECKR=(VEC1R(1,I)*VEC2R(1,I)+VEC1R(2,I)*
1 VEC2R(2,I)+VEC1R(3,I)*VEC2R(3,I))/(VR1MAG(I))
CHEK=DABS(1.0-CHECKR)
IF (CHEK .LT. 0.000000001) THEN
TEANGR(I)=0.0
ELSE
TEANGR(I)=DACOS(CHECKR)
END IF
C WRITE(9,*)'TE ANGLE V1-V2 ON RIGHT',TEANGR(I)*180./PI
C WRITE(9,*)' '
C
C CHECK FOR CONVERGENCE AT ALL OF THE WAKE NODES
C
IF (NCOUNT .LT. 19) THEN
IF (DABS(TEANGL(I)) .LT. COD) THEN
C WRITE(9,*) 'THROUGH LEFT ANGLE I,K',I,K
NODEL(K,I)=1
IF (DABS(TEANGR(I)) .LT. COD) THEN
C WRITE(9,*) 'THROUGH RIGHT ANGLE,I,K',I,K
NODER(K,I)=1
IF (K .EQ. NWAKE1 .AND. I .EQ. NSPAN) THEN
NSUML=0
NSUMR=0
DO 4560 II=1,NSPAN
DO 4560 KK=1,NWAKE1
NSUML=NSUML+NODEL(KK,II)
NSUMR=NSUMR+NODER(KK,II)
4560 CONTINUE
IF (NSUML .EQ. NSUM .AND. NSUMR .EQ. NSUM) THEN
WRITE(9,*)'ITER = 0'
ITER=0
TOTKI=0.0
DO 4565 II=1,NSPAN
II1=II+1
WI(II)=-(COSUMZ(1,II)+COSUMZ(1,II1))/2.0
WRITE(9,*)'WI (Y) ',WI(II)
4565 CONTINUE
CALL VORTEX
CALL PPITCH
CALL VPITCH
END IF
END IF
END IF
END IF
ELSE
IF (K .EQ. NWAKE1 .AND. I .EQ. NSPAN) THEN
WRITE(9,*)'ITER = 0'
WRITE(9,*)'TRUNCATED AT NCOUNT=',NCOUNT
ITER=0
TOTKI=0.0
```

```
    DO 4566 II=1,NSPAN
    II1=II+1
    WI(II)=-(COSUMZ(1,II)+COSUMZ(1,II1))/2.0
    WRITE(9,*)'WI (Y) ',WI(II)
    4566 CONTINUE
    CALL VORTEX
    CALL PPITCH
    CALL VPITCH
    END IF
    END IF
    4570 CONTINUE
    4580 CONTINUE
    END IF
    RETURN
    END
    C
    C
    C
    SUBROUTINE BIOTVEC
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    INCLUDE 'BLOK.FOR'
    C
    C THE INFLUENCE COEFFICIENTS OF THE BOUND VORTICES ARE CALCULATED
    C
    MILO=1
    DO 1200 I=1,NSPAN
    DO 1200 MATT=1,NCHORD
    QU=XVLRGT(MATT,I)
    QV=YVLRGT(MATT,I)
    QW=ZVLRGT(MATT,I)
    QX=XVLLFT(MATT,I)
    QY=YVLLFT(MATT,I)
    QZ=ZVLLFT(MATT,I)
    CALL BIGSUB
    1200 CONTINUE
    MILO=2
    C
    C THE INFLUENCE COEFFICIENTS OF THE LEFT VORTEX SEGMENTS ARE CALCU-
LATED
    C
    DO 1300 I=1,NSPAN
    DO 1300 MATT=1,NCHORD
    DO 1300 K=MATT,LINES
    K1=K+1
    IF (K .LT. NCHORD) THEN
    QU=XVLLFT(K,I)
    QV=YVLLFT(K,I)
    QW=ZVLLFT(K,I)
    QX=XVLLFT(K1,I)
    QY=YVLLFT(K1,I)
    QZ=ZVLLFT(K1,I)
    CALL BIGSUB
    END IF
    IF (K .EQ. NCHORD) THEN
    QU=XVLLFT(K,I)
    QV=YVLLFT(K,I)
    QW=ZVLLFT(K,I)
    QX=XTEL(I)
    QY=YTEL(I)
    QZ=ZTEL(I)
    CALL BIGSUB
    END IF
    IF (K .EQ. NCHORD1) THEN
    KW=K-NCHORD
    QU=XTEL(I)
    QV=YTEL(I)
    QW=ZTEL(I)
    QX=XWVLLF(KW,I)
    QY=YWVLLF(KW,I)
    QZ=ZWVLLF(KW,I)
    CALL BIGSUB
    END IF
    IF (K .GT. NCHORD1) THEN
    KW=K-NCHORD
    KW1=K-NCHORD1
    QU=XWVLLF(KW1,I)
    QV=YWVLLF(KW1,I)
    QW=ZWVLLF(KW1,I)
    QX=XWVLLF(KW,I)
    QY=YWVLLF(KW,I)
    QZ=ZWVLLF(KW,I)
    CALL BIGSUB
    END IF
    1300 CONTINUE
    C
    C THE INFLUENCE COEFFICIENTS OF THE RIGHT VORTEX SEGMENTS ARE CALCU-
LATED
    C
    DO 1400 I=1,NSPAN
    DO 1400 MATT=1,NCHORD
    DO 1400 K=MATT,LINES
    K1=K+1
    IF (K .LT. NCHORD) THEN
    QX=XVLRGT(K,I)
    QY=YVLRGT(K,I)
    QZ=ZVLRGT(K,I)
    QU=XVLRGT(K1,I)
    QV=YVLRGT(K1,I)
    QW=ZVLRGT(K1,I)
    CALL BIGSUB
    END IF
    IF (K .EQ. NCHORD) THEN
```

```
QX=XVLRGT(K,I)
QY=YVLRGT(K,I)
QZ=ZVLRGT(K,I)
QU=XTER(I)
QV=YTER(I)
QW=ZTER(I)
CALL BIGSUB
END IF
IF (K .EQ. NCHORD1) THEN
KW=K-NCHORD
QX=XTER(I)
QY=YTER(I)
QZ=ZTER(I)
QU=XWVLRT(KW,I)
QV=YWVLRT(KW,I)
QW=ZWVLRT(KW,I)
CALL BIGSUB
END IF
IF (K .GT. NCHORD1) THEN
KW=K-NCHORD
KW1=K-NCHORD1
QX=XWVLRT(KW1,I)
QY=YWVLRT(KW1,I)
QZ=ZWVLRT(KW1,I)
QU=XWVLRT(KW,I)
QV=YWVLRT(KW,I)
QW=ZWVLRT(KW,I)
CALL BIGSUB
END IF
1400 CONTINUE
RETURN
END
C
C
C
SUBROUTINE BIGSUB
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
C
C
C USING THE Q(i) VORTEX SEGMENT LOCATIONS WHICH WERE PASSED
C FROM THE MAIN PROGRAM, START A LOOP WHICH DEFINES THE
C APPROPRIATE VECTORS DEPENDING ON WHETHER THE SEGMENT IS
C LOCATED ON THE RIGHT-HALF WING (ILOC=1), THE LEFT-HALF
C WING (ILOC=2), ON THE IMAGE OF THE RIGHT-HALF WING BELOW
C THE GROUND (ILOC=3), OR ON THE IMAGE OF THE LEFT-HALF WING
C BELOW THE GROUND (ILOC=4).
C
C
PIGU=QU
PIGV=QV
```

```
PIGW=QW
PIGX=QX
PIGY=QY
PIGZ=QZ
DO 4015 ILOC=1,IGE
IF (ILOC .EQ. 2) THEN
QX=PIGU
QY=-PIGV
QZ=PIGW
QU=PIGX
QV=-PIGY
QW=PIGZ
END IF
C
C UNDERGROUND
C
IF (ILOC .EQ. 3) THEN
QY=PIGV
QV=PIGY
RADIUS=(1.0-PIGU)+SECDIS
QX=XSEC-RADIUS*DCOS(2.*ALPHA)
QZ=-PIGW-RADIUS*DSIN(2.*ALPHA)*DCOS(2.*ALPHA)
RADIUS=(1.0-PIGX)+SECDIS
QU=XSEC-RADIUS*DCOS(2.*ALPHA)
QW=-PIGZ-RADIUS*DSIN(2.*ALPHA)*DCOS(2.*ALPHA)
END IF
IF (ILOC .EQ. 4) THEN
QV=-PIGV
QY=-PIGY
RADIUS=(1.0-PIGU)+SECDIS
QU=XSEC-RADIUS*DCOS(2.*ALPHA)
QW=-PIGW-RADIUS*DSIN(2.*ALPHA)*DCOS(2.*ALPHA)
RADIUS=(1.0-PIGX)+SECDIS
QX=XSEC-RADIUS*DCOS(2.*ALPHA)
QZ=-PIGZ-RADIUS*DSIN(2.*ALPHA)*DCOS(2.*ALPHA)
END IF
RO(1)=QU-QX
RO(2)=QV-QY
RO(3)=QW-QZ
C
IF (NEVEN .EQ. 1) THEN
DO 4000 J=1,NSPAN
DO 4000 L=1,NCHORD
R1(1)=XCP(L,J)-QX
R1(2)=YCP(L,J)-QY
R1(3)=ZCP(L,J)-QZ
R2(1)=XCP(L,J)-QU
R2(2)=YCP(L,J)-QV
R2(3)=ZCP(L,J)-QW
IF (MILO .EQ. 1) THEN
IF (ILOC .EQ. 1) THEN
```

```
COEFX(L,J,MATT,I)=0.0
COEFY(L,J,MATT,I)=0.0
COEFZ(L,J,MATT,I)=0.0
END IF
END IF
C
CALL CROSS(R1,R2,RCROSS)
CALL SMAG(R1,R2,RCROSS,R1MAG,R2MAG,RCRMAG)
CALL DOT(RO,R1,R2,DOTRO1,DOTRO2)
CALL FACTOR(RCROSS,RCRMAG,FAC1)
IF (DABS(R1MAG) .GT. .000000001) THEN
FAC2=DOTRO1/R1MAG
ELSE
FAC2=0.0
END IF
IF (DABS(R2MAG) .GT. .000000001) THEN
FAC2=FAC2-DOTRO2/R2MAG
END IF
C
COEFX(L,J,MATT,I)=.25*FAC2*FAC1(1)*DCOS(DI)*
1 DSIN(DELTD(L,J))/PI+COEFX(L,J,MATT,I)
COEFY(L,J,MATT,I)=.25*FAC2*FAC1(2)*DSIN(DI)*
1 DCOS(DELTD(L,J))/PI+COEFY(L,J,MATT,I)
COEFZ(L,J,MATT,I)=-.25*FAC2*FAC1(3)*DCOS(DI)*
1 DCOS(DELTD(L,J))/PI+COEFZ(L,J,MATT,I)
4000 CONTINUE
ELSE
DO 4010 J=1,NSPAN1
DO 4010 L=1,NWAKE1
JM1=J-1
LM1=L-1
IF (J .LT. NSPAN1) THEN
IF (L .EQ. 1) THEN
R1(1)=XTEL(J)-QX
R1(2)=YTEL(J)-QY
R1(3)=ZTEL(J)-QZ
R2(1)=XTEL(J)-QU
R2(2)=YTEL(J)-QV
R2(3)=ZTEL(J)-QW
ELSE
R1(1)=XWVLLF(LM1,J)-QX
R1(2)=YWVLLF(LM1,J)-QY
R1(3)=ZWVLLF(LM1,J)-QZ
R2(1)=XWVLLF(LM1,J)-QU
R2(2)=YWVLLF(LM1,J)-QV
R2(3)=ZWVLLF(LM1,J)-QW
END IF
IF (MILO .EQ. 1) THEN
IF (ILOC .EQ. 1) THEN
COSUMX(L,J)=0.0
COSUMY(L,J)=0.0
```

```
COSUMZ(L,J)=0.0
END IF
END IF
CALL CROSS(R1,R2,RCROSS)
CALL SMAG(R1,R2,RCROSS,R1MAG,R2MAG,RCRMAG)
CALL DOT(RO,R1,R2,DOTRO1,DOTRO2)
CALL FACTOR(RCROSS,RCRMAG,FAC1)
IF (DABS(R1MAG) .GT. .0000001) THEN
FAC2=DOTRO1/R1MAG
ELSE
FAC2=0.0
END IF
IF (DABS(R2MAG) .GT. .0000001) THEN
FAC2=FAC2-DOTRO2/R2MAG
END IF
C
COSUMX(L,J)=.25*FAC2*FAC1(1)*GAMMA(MATT,I)/PI+
1 COSUMX(L,J)
COSUMY(L,J)=.25*FAC2*FAC1(2)*GAMMA(MATT,I)/PI+
1 COSUMY(L,J)
COSUMZ(L,J)=.25*FAC2*FAC1(3)*GAMMA(MATT,I)/PI+
1 COSUMZ(L,J)
C
C IF THE FAR RIGHT NODE IS EVALUATED
C
ELSE
IF (L .EQ. 1) THEN
R1(1)=XTER(JM1)-QX
R1(2)=YTER(JM1)-QY
R1(3)=ZTER(JM1)-QZ
R2(1)=XTER(JM1)-QU
R2(2)=YTER(JM1)-QV
R2(3)=ZTER(JM1)-QW
ELSE
R1(1)=XWVLRT(LM1,JM1)-QX
R1(2)=YWVLRT(LM1,JM1)-QY
R1(3)=ZWVLRT(LM1,JM1)-QZ
R2(1)=XWVLRT(LM1,JM1)-QU
R2(2)=YWVLRT(LM1,JM1)-QV
R2(3)=ZWVLRT(LM1,JM1)-QW
END IF
IF (MILO .EQ. 1) THEN
IF (ILOC .EQ. 1) THEN
COSUMX(L,J)=0.0
COSUMY(L,J)=0.0
COSUMZ(L,J)=0.0
END IF
END IF
C
CALL CROSS(R1,R2,RCROSS)
CALL SMAG(R1,R2,RCROSS,R1MAG,R2MAG,RCRMAG)
CALL DOT(RO,R1,R2,DOTRO1,DOTRO2)
CALL FACTOR(RCROSS,RCRMAG,FAC1)
IF (DABS(R1MAG) .GT. .0000001) THEN
FAC2=DOTRO1/R1MAG
ELSE
FAC2=0.0
END IF
IF (DABS(R2MAG) .GT. .0000001) THEN
FAC2=FAC2-DOTRO2/R2MAG
END IF
C
COSUMX(L,J)=.25*FAC2*FAC1(1)*GAMMA(MATT,I)/PI+
1 COSUMX(L,J)
COSUMY(L,J)=.25*FAC2*FAC1(2)*GAMMA(MATT,I)/PI+
1 COSUMY(L,J)
COSUMZ(L,J)=.25*FAC2*FAC1(3)*GAMMA(MATT,I)/PI+
1 COSUMZ(L,J)
END IF
4010 CONTINUE
END IF
4015 CONTINUE
RETURN
END
C
C
C
SUBROUTINE CROSS(R1,R2,RCROSS)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION R1(3),R2(3),RCROSS(3)
RCROSS(1)=R1(2)*R2(3)-R2(2)*R1(3)
RCROSS(2)=R2(1)*R1(3)-R1(1)*R2(3)
RCROSS(3)=R1(1)*R2(2)-R2(1)*R1(2)
RETURN
END
C
C
C
SUBROUTINE SMAG(R1,R2,RCROSS,R1MAG,R2MAG,RCRMAG)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION R1(3),R2(3),RCROSS(3)
R1MAG=(R1(1)**2+R1(2)**2+R1(3)**2)**.5
R2MAG=(R2(1)**2+R2(2)**2+R2(3)**2)**.5
RCRMAG=(RCROSS(1)**2+RCROSS(2)**2+RCROSS(3)**2)**.5
RETURN
END
C
C
C
SUBROUTINE DOT(RO,R1,R2,DOTRO1,DOTRO2)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION R1(3),R2(3),RO(3)
```

```
DOTRO1=RO(1)*R1(1)+RO(2)*R1(2)+RO(3)*R1(3)
DOTRO2=RO(1)*R2(1)+RO(2)*R2(2)+RO(3)*R2(3)
RETURN
END
C
C
C
SUBROUTINE FACTOR(RCROSS,RCRMAG,FAC1)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION RCROSS(3),FAC1(3)
IF (RCRMAG .GT. .00000001) THEN
FAC1(1)=RCROSS(1)/RCRMAG**2
FAC1(2)=RCROSS(2)/RCRMAG**2
FAC1(3)=RCROSS(3)/RCRMAG**2
ELSE
FAC1(1)=0.0
FAC1(2)=0.0
FAC1(3)=0.0
END IF
RETURN
END
C
C
C
SUBROUTINE CNDNS4
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
MOO=0
LOO=MOO
DO 4200 I=1,NSPAN
DO 4200 K=1,NCHORD
MOO=MOO+1
DO 4100 J=1,NSPAN
DO 4100 L=1,NCHORD
LOO=LOO+1
COEFX2(LOO,MOO)=COEFX(L,J,K,I)
COEFY2(LOO,MOO)=COEFY(L,J,K,I)
COEFZ2(LOO,MOO)=COEFZ(L,J,K,I)
A(LOO,MOO)=COEFX2(LOO,MOO)+COEFY2(LOO,MOO)+
1 COEFZ2(LOO,MOO)
4100 CONTINUE
LOO=0
4200 CONTINUE
RETURN
END
C
C
C
SUBROUTINE CNDNS2
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
```

```
IO=0
DO 4300 J=1,NSPAN
DO 4300 L=1,NCHORD
IO=IO+1
DELTS(IO)=DATAN(DZDX(L,J))
DELTD(L,J)=DELTS(IO)
4300 CONTINUE
RETURN
END
C
C
C
SUBROUTINE WLIFT
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
CLIFT=0.0
TOTKP=0.0
LORI=0
DO 620 J=1,NSPAN
CLISPN=0.0
DO 610 L=1,NCHORD
LORI=LORI+1
GAMMA(L,J)=B(LORI)
CLIFT=GAMMA(L,J)*DCOS(ALPHA)*DCOS(ALPHA)+CLIFT
CLISPN=GAMMA(L,J)+CLISPN
610 CONTINUE
YKAPE(J)=CLISPN*WEIGHT/(DSIN(ALPHA))
TOTKP=YKAPE(J)+TOTKP
620 CONTINUE
CLIFT=CLIFT*WEIGHT
RETURN
END
C
C
C
SUBROUTINE VORTEX
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
CSUCK=0.0
LLAMA=0
DO 720 J=1,NSPAN
CLISPN=0.0
DO 710 L=1,NCHORD
LLAMA=LLAMA+1
GAMMA(L,J)=B(LLAMA)
CLISPN=GAMMA(L,J)+CLISPN
710 CONTINUE
CORE=YCP(1,J)
CALL FOUND
CONVRT=DCOS(ALPHA)/DSIN(XWEE)
CSUCK=CLISPN*CONVRT*(DSIN(ALPHA)-WI(J))+CSUCK
```

```
720 CONTINUE
CLVORT=CSUCK*WEIGHT
WRITE(6,*)'BCROP,BPANEL',BCROP,BPANEL
TOTKV=CLVORT/(DSIN(ALPHA)*DSIN(ALPHA)*DCOS(ALPHA))
WRITE(6,*)'WEIGHT',WEIGHT
WRITE(6,*)'DELY',DELY
RETURN
END
C
C
C
SUBROUTINE PPITCH
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
PPIT=0.0
LORI=0
DO 620 J=1,NSPAN
DO 610 L=1,NCHORD
LORI=LORI+1
GAMMA(L,J)=B(LORI)
PPIT=GAMMA(L,J)*DCOS(ALPHA)*(CMOM-XCP(L,J))+PPIT
610 CONTINUE
620 CONTINUE
PPIT=PPIT*WEIGHT
RETURN
END
C
C
C
SUBROUTINE VPITCH
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
CSUCK=0.0
VPIT=0.0
LLAMA=0
DO 720 J=1,NSPAN
CLISPN=0.0
DO 710 L=1,NCHORD
LLAMA=LLAMA+1
GAMMA(L,J)=B(LLAMA)
CLISPN=GAMMA(L,J)+CLISPN
710 CONTINUE
CORE=YCP(1,J)
CALL FOUND
CONVRT=DCOS(ALPHA)/DSIN(XWEE)
TEMVIT=CLISPN*CONVRT*(DSIN(ALPHA)-WI(J))
VPIT=TEMVIT*(CMOM-ALOSS)/CBAR+VPIT
720 CONTINUE
VPIT=VPIT*WEIGHT
PITMOM=VPIT+PPIT
CALL OUT
```

```
RETURN
END
C
C
C
SUBROUTINE OUT
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INCLUDE 'BLOK.FOR'
C
WRITE(9,13)'ANGLE OF ATTACK',ALPHA*180./PI
WRITE(9,11)'ASPECT RATIO',AR
WRITE(9,11)'TAPER RATIO, CT/CO',CT
WRITE(9,11)'NOTCH RATIO, CN/CO',CN
DO 8000 JM=1,LEEDS
WRITE(9,11)'LEADING EDGE SWEEP',90.-SWEEP(JM)*180./PI
8000 CONTINUE
WRITE(9,13)'POTENTIAL LIFT COEFFICIENT',CLIFT
WRITE(9,11)'VORTEX LIFT COEFFICIENT',CLVORT
WRITE(9,12)'TOTAL LIFT COEFFICIENT',CLVORT+CLIFT
C
WRITE(9,13)'POTENTIAL PITCHING MOMENT',PPIT
WRITE(9,11)'VORTEX PITCHING MOMENT',VPIT
WRITE(9,12)'TOTAL PITCHING MOMENT',PITMOM
C
WRITE(9,13)'POTENTIAL CONSTANT, KP',TOTKP
WRITE(9,11)'VORTEX CONSTANT, KV',TOTKV
C
11 FORMAT(3X,A27,5X,F9.4)
12 FORMAT(/,3X,A27,5X,F9.4)
13 FORMAT(////3X,A27,5X,F9.4)
RETURN
END
```

Filename: BLOK.FOR

```
C
COMMON NSPAN,NCHORD,QU,QV,QW,QX,QY,QZ,COEFX,COEFY,COEFZ,
1 XCP,YCP,ZCP,XWCP,YWCP,ZWCP,MATT,I,MILO,NWAKE,WAKDEL,XTEL,YTEL,
2 ZTEL,XTER,YTER,ZTER,XWVLLF,YWVLLF,ZWVLLF,XWVLRT,YWVLRT,ZWVLRT,
3 NCOUNT,NSPAN1,PI,NEVEN,ITER,COSUMX,COSUMY,COSUMZ,NCHORD1,DI,
4 NWAKE1,DELY,LINES,GAMMA,K,UINF,CO,CN,CT,A,B,DELTS,DELTD,CLIFT,
5 ALPHA,ILOC,IGE,HOVB,WEIGHT,CRDATY,BCROP,XVLLFT,XVLRGT,YVLLFT
COMMON YVLRGT,ZVLLFT,ZVLRGT,KAU,KAOI,IER,IDGT,M,N,IA,LESYM,LOW,
1 FLINX,FLINY,FOUTX,FOUTY,PERX,PERY,XPOIN,YPOIN,ZPOIN,FUNPTX,
2 FUNPTY,FANG,JA,DELLE,IDFLECT,NUML,NLEFL,ILIGN,FLASWE,NUMEDGL,
3 ITESYM,ITEFLEC,TOUTX,TOUTY,TFINX,TFINY,TFLSWE,TFANG,ILINTI,
4 ILINTO,NTEFL,DELTE,LOT,TOX,TOY,NUMT,SPOCK,CLVORT,WI,LEEDM1,
5 CRAN,LEEDS,NTREDS,TWEEP,TRCRA,SWEEP,DELYB,CORDCL,YCPB
COMMON CORE,ALOSS,TRSPN,NTREM1,EXTRA,XWEE,
1 TOTKP,TOTKV,TOTKI,YKAPE,YKAVE,YKI,AIREA,BPANEL,
2 VPIT,PPIT,PITMOM,CNORML,CBAR,CMOM,SECDIS,XSEC,DZDX
DIMENSION RO(3),R1(3),R2(3),RCROSS(3),FAC1(3),DELTS(100),
1 DELTD(10,10),B(100),XTEL(10),CRDATY(10),CORDCL(10),YTEL(10),
2 ZTEL(10),XTER(10),YTER(10),ZTER(10),XCP(10,10),YCP(10,10),
3 YCPB(10,10),ZCP(10,10),XWCP(20,10),YWCP(20,10),ZWCP(20,10),
4 XVLLFT(10,10),YVLLFT(10,10),ZVLLFT(10,10),XVLRGT(10,10),
5 YVLRGT(10,10),ZVLRGT(10,10),A(100,100),XWVLLF(20,10),
6 YWVLLF(20,10),ZWVLLF(20,10),XWVLRT(20,10),YWVLRT(20,10)
DIMENSION ZWVLRT(20,10),COEFX2(100,100),COEFY2(100,100),
1 COEFZ2(100,100),COEFX(10,10,10,10),COEFY(10,10,10,10),
2 COEFZ(10,10,10,10),DZDX(10,10),GAMMA(10,10),COSUMX(20,10),
3 COSUMY(20,10),COSUMZ(20,10),WKAREA(100),CO(10),SWEEP(10),
4 CRAN(9),ADAR(9),TWEEP(10),TRCRA(9),TRSPN(9),
5 FANG(10),FLASWE(10),DELLE(10),NUML(10),IDFLECT(10),FLINX(10),
6 FLINY(10),FOUTX(10),FOUTY(10),ITESYM(10),TFANG(10),TFLSWE(10)
DIMENSION DELTE(10),NUMT(10),ITEFLEC(10),TFINX(10),TFINY(10),
1 TOUTX(10),TOUTY(10),ILINTI(10),ILINTO(10),WI(10),YKAPE(10),
2 YKAVE(10),YKI(10)
```

Filename: PANEL.FOR

```
C
C WING IS 9CH X 9SP MAX AND WAKE IS 19CH X 9SP MAX
C
NCHORD=5
NSPAN=4
NWAKE=10
WAKDEL=.20
```

Filename: DATA.FOR

```
C
ALPHA=(PI/180.)*24.
C IGE=2
IGE=4
HOVB=.1
DI=-.10*(PI/180.)
CO(1)=1.0
CT=0.0
CN=0.0
C CBAR=(2./3.)*CO(1)
CBAR=CO(1)
C CMOM=CO(1)/2.
CMOM=0.0
C
LEEDS=1
C SWEEP(1)=(PI/180.)*(90.0-45.000)
C SWEEP(1)=(PI/180.)*(90.0-48.814)
C SWEEP(1)=(PI/180.)*(90.0-53.1301)
C SWEEP(1)=(PI/180.)*(90.0-57.995)
C SWEEP(1)=(PI/180.)*(90.0-63.435)
C SWEEP(1)=(PI/180.)*(90.0-69.444)
C SWEEP(1)=(PI/180.)*(90.0-75.9368)
SWEEP(1)=(PI/180.)*(90.0-70.0)
C SWEEP(1)=(PI/180.)*(90.0-89.0)
```

```
C
NTREDS=1
TWEEP(1)=(PI/180.)*0.
SPOCK=.20 !!!SINCE TWEEP<.001!!!ALWAYS NEED THIS??
C
NLEFL=0
C
NTEFL=1
ILINTI(1)=1
ILINTO(1)=5
TFANG(1)=0.0
TFLSWE(1)=0.0
TFINX(1)=1.0
TFINY(1)=0.0
TOUTX(1)=.88
TOUTY(1)=.3203
ITEFLEC(1)=1
DELTE(1)=(PI/180.)*8.5
C
NUMEDGL=1
```

Nomenclature

A, B, C = points representing a set of (X,Y,Z) coordinates

AR = aspect ratio

C_L = wing lift coefficient

C_M = wing pitching moment coefficient

C_{M_o} = wing pitching moment coefficient at $C_L = 0$

C_N = wing normal force coefficient

C_S = wing suction coefficient

C_T = wing thrust coefficient

C_l = section lift coefficient

C_m = section moment coefficient

C_{mn} = influence coefficient at m due to n

C_n = section normal force coefficient

C_s = section suction coefficient

C_t = section thrust coefficient

DIST = x-direction distance between x_2 and INT′

IGE = in ground effect

INT = distance between the points $x = 1$ and $x =$ INT′

INT′ = x-coordinate of the intersection of the rearward extension of the wing and the image plane

K_p = suction analogy potential constant

K_v = suction analogy vortex constant

L = wing lift

NCHORD = number of panels on the chord

NSPAN = number of panels on the semi-span

OGE = out of ground effect

P = pressure

S = wing area

T = transformation

V = free-stream velocity

XTE(I) = x-coordinate of a particular trailing-edge point

X, Y, Z = spatial coordinates

b = wing span

c = section chord

dl = incremental lift force

$\overrightarrow{dl}$ = incremental distance in the direction of the orientation of the axis of a vortex filament

$\overrightarrow{dV}$ = incremental induced velocity

dx, dy, dz = incremental distance

h = height above the ground

h/b = non-dimensional height above the ground

l = airfoil lift

le = leading edge

$\overrightarrow{r}$ = vector which connects an endpoint of a vortex segment to another vector

te = trailing edge

u, v, w = velocity components in the x-, y-, and z-directions, respectively

x, y, z = spatial coordinates

Γ = total circulation strength for a wing

Δwake = distance between successive nodes in the wake

Δx = distance that the wake is moved in the x-direction

Δy = width of the panels representing the wing, or the distance that the wake is moved in the y-direction

Δz = distance that the wake is moved in the z-direction

θ = angular measure

Λ = leading-edge sweep angle, measured from the root-chord line to the leading-edge line

ϕ	= angular measure
α	= angle of attack
γ	= circulation strength for an airfoil or vortex
δ	= flap deflection
ρ	= density of the air

Subscripts

D	= distributed over one spatial dimension
REF	= reference value used in a "source"
source	= referring to another's work
T	= trailing edge related
e	= elevon
i	= summation subscript or an individual item
in	= induced
j	= summation subscript
n	= an individual item
o	= root chord
p	= perpendicular (for vectors)
0, 1, 2, 3	= referring to similar points at different locations or at different times.

Superscripts

—	= reference value
˙	= time rate of change