Not in

AD 637872

TECHNICAL REPORT

THEORETICAL PREDICTION OF THE FLOW IN THE WAKE OF A HELICOPTER ROTOR

ADDENDUM - EFFECTS DUE TO A FUSELAGE IN A CONSTANT, NONUNIFORM FLOW

By: Peter Crimi and Andrew R. Trenka

CAL No. BB-1994-S-3

Prepared for:
U.S. Army
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland 21005

Final Report - Addendum
Contract No. DA30-069-AMC-645(R)
August 1966

CORNELL AERONAUTICAL LABORATORY, INC.
OF CORNELL UNIVERSITY, BUFFALO, N. Y. 14221

CORNELL AERONAUTICAL LABORATORY, INC.
BUFFALO, NEW YORK 14221

THEORETICAL PREDICTION OF THE FLOW IN THE WAKE OF A HELICOPTER ROTOR

ADDENDUM

EFFECTS DUE TO A FUSELAGE IN A CONSTANT, NONUNIFORM FLOW

By:

PETER CRIMI
ANDREW R. TRENKA

CAL NO. BB-1994-S-3

FINAL REPORT - ADDENDUM

CONTRACT NO. DA30-069-AMC-645(R)

AUGUST 1966

PREPARED FOR

U.S ARMY
BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND 21005

SUMMARY

This addendum reports the results of an extension to the study carried out at Cornell Aeronautical Laboratory for the U. S. Army (Contract No. DA 30-069-AMC-645(R)) to determine the time varying flow in the vicinity of a helicopter rotor in hovering or forward flight and having a fuselage immersed in the rotor wake. The purpose of this extension was to compare two models representing the fuselage. The first fuselage model, reported in Parts 1 and 2, was based upon the assumption that the fuselage was immersed in a constant and uniform flow; the second model was based upon the assumption that the fuselage was immersed in a constant but non-uniform flow.

Comparisons between the two models are presented.

Also presented is the digital program employed to solve the problem.

FOREWORD

The work reported herein, performed between January 1966 and April 1966, was accomplished by the Cornell Aeronautical Laboratory, Inc. (CAL), Buffalo, New York, for the Director of Ballistic Research Laboratories, (BRL) Aberdeen Proving Ground, Maryland. The research effort was performed under Contract DA 30-069-AMC-645(R) and was monitored for BRL by Mr. Thomas Coyle as Technical Supervisor. Dr. Peter Crimi and Mr. Andrew R. Trenka of CAL conducted the study. Mr. Harvey Selib developed the digital computer program.

This document is an addendum to Parts 1 and 2 of the final report under the contract. It describes the modifications made to the theory and the digital program to allow the determination of the effects of a fuselage immersed in a constant, nonuniform flow. Part 1 of the final report describes the development of the theory, discusses the results of the computation, and provides a comprehensive discussion of the work performed under the contract. Part 2 of the final report describes the formulation and application of the rotor wake-flow computer program and is of use primarily to those who plan to use the digital computing program.

CAL Report Numbers have been assigned as follows:

BB-1994-S-1, THEORETICAL PREDICTION OF THE FLOW IN THE WAKE OF A HELICOPTER ROTOR, Part 1 - Development of Theory and Results of Computations

BB-1994-S-2, THEORETICAL PREDICTION OF THE FLOW IN THE WAKE OF A HELICOPTER ROTOR, Part 2 - Formulation and Application of the Rotor Wake Flow Computer Programs

BB-1994-S-3, THEORETICAL PREDICTION OF THE FLOW IN THE WAKE OF A HELICOPTER ROTOR, Addendum - Effects Due to a Fuselage in a Constant, Nonuniform Flow

CONTENTS

ILLUSTRATIONS

SYMBOLS

B_{mn}	coefficients relating velocities induced by element n and its image on element m
N_A	number of azimuthal stations per revolution
N_B	number of rotor blades
N_f	number of fuselage points
R	rotor radius
V_f	free stream velocity
$V_x^{(m)}$ & $V_z^{(m)}$	x and z components of the time averaged local velocity at m
V_{x_∞} & V_{z_∞}	time and spacial averaged velocities in the x and z directions
$V_{x_f}, V_{y_f}, V_{z_f}$	(x, y, z) components of velocity induced by the fuselage
$V_{x_m}, V_{y_m}, V_{z_m}$	the (x, y, z) components of velocity induced at a point m by a source of unit strength
W	aircraft weight
x, y, z	rectilinear coordinates with origin in the tip-path plane, nondimensionalized by R
α_T	inclination of the tip-path plane to the free stream
λ	blade loading parameter; $\lambda = 4W/(\pi^2 N_B \rho \Omega^2 R^4)$
λ_{ξ_m} & ν_{ξ_m}	direction cosines of the normal to the m^{th} element with respect to the (x, z) axes
μ	advance ratio; $\mu = V_f/\Omega R$
ρ	air density
σ_n	normal component of the normalized source strength per unit area of the n^{th} element
σ_{xn}	x-component of the normalized source strength per unit area of the n^{th} element

σ_{zn} z-component of the normalized source strength per unit area of the n^{th} element

ψ_k rotor azimuth angle at k^{th} position

Ω rotor ang ılar speed

1. INTRODUCTION

As was described in Part 1 of this report (Reference 1), a model for a helicopter fuselage was developed consisting of an array of quadrilateral source sheets. This model formed part of the representation used for determining the flow in the wake of a helicopter rotor. The strengths of the individual source sheets were assigned by satisfying the fuselage boundary condition resulting from a steady uniform free stream. The magnitude and direction of this free stream were determined by computing an average in space and time of the flow obtained, in the absence of the fuselage representation, in the region the fuselage would normally occupy.

Extensive calculations revealed that the fuselage was adequately represented in this manner for a wide range of flight conditions, provided the region of interest in the flow was sufficiently removed (on the order of one to two-tenths of a rotor radius) from the fuselage. However, considerable spacial variation of the time average of the flow over the region occupied by the fuselage was observed to be present in most cases, so that the flow was not well represented in the immediate vicinity of the fuselage surface.

This addendum reports the results of work performed subsequent to completion of the study reported in References 1 and 2. The objective of this continuation was to develop and evaluate an improved fuselage representation which takes account of the nonuniformity of the time average of the flow imposed on the fuselage. The procedures used in implementing the improved model are described in Section 2. The results of the computations performed are discussed in Section 3. A listing of the modified digital programs is presented in Appendixes I and II. Familiarity with the developments of References 1 and 2 is assumed in these descriptions and discussions.

2. PROCEDURES USED TO ACCOUNT FOR NONUNIFORM FLOW OVER THE FUSELAGE

To account for the nonuniformity of the time-averaged flow over the fuselage, it was necessary to make changes both in the main rotor-wake-flow program and in the supplemental fuselage program. The changes were coded as alternative procedures; so the original structures of the programs were retained. The program changes will be outlined first, and their significance and implementation will then be discussed.

Consider first the revisions to the supplemental fuselage program. The supplemental fuselage program which determines the normalized source strengths was modified to solve for the source strengths σ_n, in a specified nonuniform flow field. The set of equations (Equations (20) of Reference 2)

$$\left.\begin{aligned} \sum_{n=1}^{N_f} B_{mn}\,\sigma_{x_n} &= -\lambda_{\zeta_m} \\ \sum_{n=1}^{N_f} B_{mn}\,\sigma_{z_n} &= -\nu_{\zeta_m} \end{aligned}\right\} ; \quad m = 1, 2, \cdots N_f$$

was replaced by the set

$$\sum_{n=1}^{N_f} B_{mn}\,\sigma_n = -\left[\lambda_{\zeta_m} V_x^{(m)} + \nu_{\zeta_m} V_z^{(m)}\right]; \quad m = 1, 2, \cdots N_f \qquad (20a)$$

where

$V_x^{(m)}$ & $V_z^{(m)}$	are known x and z components of the time-averaged local velocity at the m^{th} element
λ_{ζ_m} & ν_{ζ_m}	are the direction cosines of the normal to the m^{th} element with respect to the (x, z) axes.
B_{mn}	are coefficients relating velocities induced by element n and its image on element m
N_f	is the number of fuselage points.

All other relationships of the fuselage program are unchanged.

The main program which computed the position of the wake was modified to account for the contribution of a fuselage in a steady nonuniform flow as follows. The equations (Equations (13) of Reference 2)

$$
\begin{aligned}
V_{x_f}(x,y,z) &= \sum_{m=1}^{N_f} \left[\sigma_{x_m} V_{x_\infty} + \sigma_{z_m} V_{z_\infty}\right]\left[V_{x_m} + \tilde{V}_{x_m}\right] \\
V_{y_f}(x,y,z) &= \sum_{m=1}^{N_f} \left[\sigma_{x_m} V_{x_\infty} + \sigma_{z_m} V_{z_\infty}\right]\left[V_{y_m} - \tilde{V}_{y_m}\right] \qquad (13) \\
V_{z_f}(x,y,z) &= \sum_{m=1}^{N_f} \left[\sigma_{x_m} V_{x_\infty} + \sigma_{z_m} V_{z_\infty}\right]\left[V_{z_m} + \tilde{V}_{z_m}\right],
\end{aligned}
$$

which define x, y, z components of the velocity induced by the fuselage at a point (x, y, z), were replaced by the equations

$$
\left.
\begin{aligned}
V_{x_f}(x,y,z) &= \sum_{m=1}^{N_f} \sigma_m \left(V_{x_m} + \tilde{V}_{x_m}\right) \\
V_{y_f}(x,y,z) &= \sum_{m=1}^{N_f} \sigma_m \left(V_{y_m} - \tilde{V}_{y_m}\right) \\
V_{z_f}(x,y,z) &= \sum_{m=1}^{N_f} \sigma_m \left(V_{z_m} - \tilde{V}_{z_m}\right)
\end{aligned}
\right\} \qquad (13a)
$$

where

V_{x_∞} & V_{z_∞} are the time and spacial averaged velocities in the x and z directions.

V_{x_m}, V_{y_m}, V_{z_m} are the (x, y, z) components of velocity induced at a point by a source of unit strength at the m^{th} element (the tilde (~) indicates the velocity induced by the image of the element).

That is, the quantity $[\sigma_{x_m} V_{x_\infty} + \sigma_{z_m} V_{z_\infty}]$ is simply replaced by σ_m as computed by solving Equations (20a).

The modified programs are utilized in the following manner. First a flight condition is chosen, by specifying μ, λ, α_T, etc. Then, the main program is run with the fuselage representation omitted, until periodicity is established in the wake flow. The computations are then continued through one period (i.e. through N_A/N_B azimuth positions), computing the flow at the points

$$(\bar{x}_m, \bar{y}_m, \bar{z}_m), (\bar{x}_m, -\bar{y}_m, \bar{z}_m), \qquad m = 1, 2, \cdots, N_f .$$

It will be recalled that these are the points and their images with respect to the fuselage plane of symmetry at which the fuselage boundary condition is satisfied. Nonuniformity in the flow is taken into account by computing a time average of the flow at each of these points, and then requiring the fuselage source strengths to be such as to just cancel the component of this average flow which is normal to the element. Specifically, the quantities $V_x^{(m)}$ and $V_z^{(m)}$ of Equations (20a) are computed according to

$$V_x^{(m)} = \frac{N_B}{2N_A} \left\{ \sum_{k=1}^{N_A/N_B} \left[V_x(\bar{x}_m, \bar{y}_m, \bar{z}_m ; \psi_k) + V_x(\bar{x}_m, -\bar{y}_m, \bar{z}_m ; \psi_k) \right] \right\}$$

$$V_z^{(m)} = \frac{N_B}{2N_A} \left\{ \sum_{k=1}^{N_A/N_B} \left[V_z(\bar{x}_m, \bar{y}_m, \bar{z}_m ; \psi_k) + V_z(\bar{x}_m, -\bar{y}_m, \bar{z}_m ; \psi_k) \right] \right\}$$

where V_x and V_z are components of fluid velocity in the x and z directions, respectively, as defined by Equations (3) of Reference 2, and ψ_k denotes the k^{th} rotor azimuth position. Note that the flow at the image point has been averaged into $V_x^{(m)}$ and $V_z^{(m)}$ to account in part for any asymmetry in the flow with respect to the x-z plane.

The values of $V_x^{(m)}$ and $V_z^{(m)}$ as so computed are supplied as inputs to the fuselage program and utilized in Equation (20a). The right-hand side of that equation may be identified as the negative of the component of the time-averaged velocity at $(\bar{x}_m, \bar{y}_m, \bar{z}_m)$ which is normal to element m. The source strengths σ_m are computed by solving Equations (20a) and are then made available to the main program where they are used in Equations (13a) when computing the flow at a given point.

3. RESULTS OF COMPUTATIONS WHICH ACCOUNT FOR NONUNIFORM FLOW OVER THE FUSELAGE

GENERAL REMARKS

It was reasoned in Reference 1 that the assumption of uniform flow employed to determine the strengths of the source sheets representing the fuselage was most questionable in the region close to the fuselage when the rotor was in the flight range $0.05 \leq \mu \leq 0.10$. In this flight range, wake vortices are swept directly over the region which is occupied by the fuselage. Thus spacial variations in the velocity occur which are much larger than those experienced at higher or lower advance ratios. Since the fuselage influence is substantial in this range of advance ratios, these large spacial variations should be important in determining the contribution of the fuselage to the net flow. To determine the soundness of this reasoning, the analysis of two flight configurations reported in Reference 1, which had been based on a constant and uniform flow over the fuselage, were repeated taking account of the nonuniform flow over the fuselage. The results of the computations are discussed below. To facilitate comparisons between the two fuselage models, the results are presented for the same spacial locations as in Reference 1.

THE FLOW AT A HIGH ADVANCE RATIO

The flight configuration at an advance ratio of 0.220 was selected as typical of the high advance ratio flows. This case was computed using a nonuniform flow to determine the fuselage source strengths. Presented in Figures 1 through 4 are plots of the spacial distributions of the velocity components at various rotor azimuths. Comparisons are shown between the velocities computed using the uniform and nonuniform fuselage flow models.

As expected, no appreciable difference was obtained in the velocities computed at distances greater than 0.2R from the fuselage. The largest deviations were obtained in the V_x and V_z components in the immediate vicinity of the fuselage (see Figure 1). The computed value of the V_z component using the nonuniform flow model was found to be reduced by approximately 10%, while the V_x component was increased by approximately 10%.

The spacial variations in the velocities are still primarily attributable to the fuselage and the azimuthal variations remained nearly constant.

THE FLOW AT A LOW ADVANCE RATIO

Of the cases investigated in Reference 1, the flow at $\mu = 0.0732$ was found to exhibit a relatively large dependence upon the fuselage (see Figure 46 of Reference 1). It was also found that the spacial variation of the downwash in the volume occupied by the fuselage (computed with the fuselage removed from the flow) was large (see Figure 55 of Reference 1). Hence, this case was selected to be recomputed using the nonuniform flow model to determine the source strengths representing the fuselage. The results of the computations are presented in Figures 5 through 10 along with the corresponding results for the uniform flow model.

It was found that all of the velocity components were altered. The magnitudes of the differences and whether the variation was an increase or a decrease depended upon the distances from the fuselage and the rotor plane. The velocities computed using the nonuniform flow model differed from the velocities computed using the uniform flow model by 8% to 14% in the region 0.1R to 0.2R from the fuselage.

The general character of the flow was found, however, to be the same for both models.

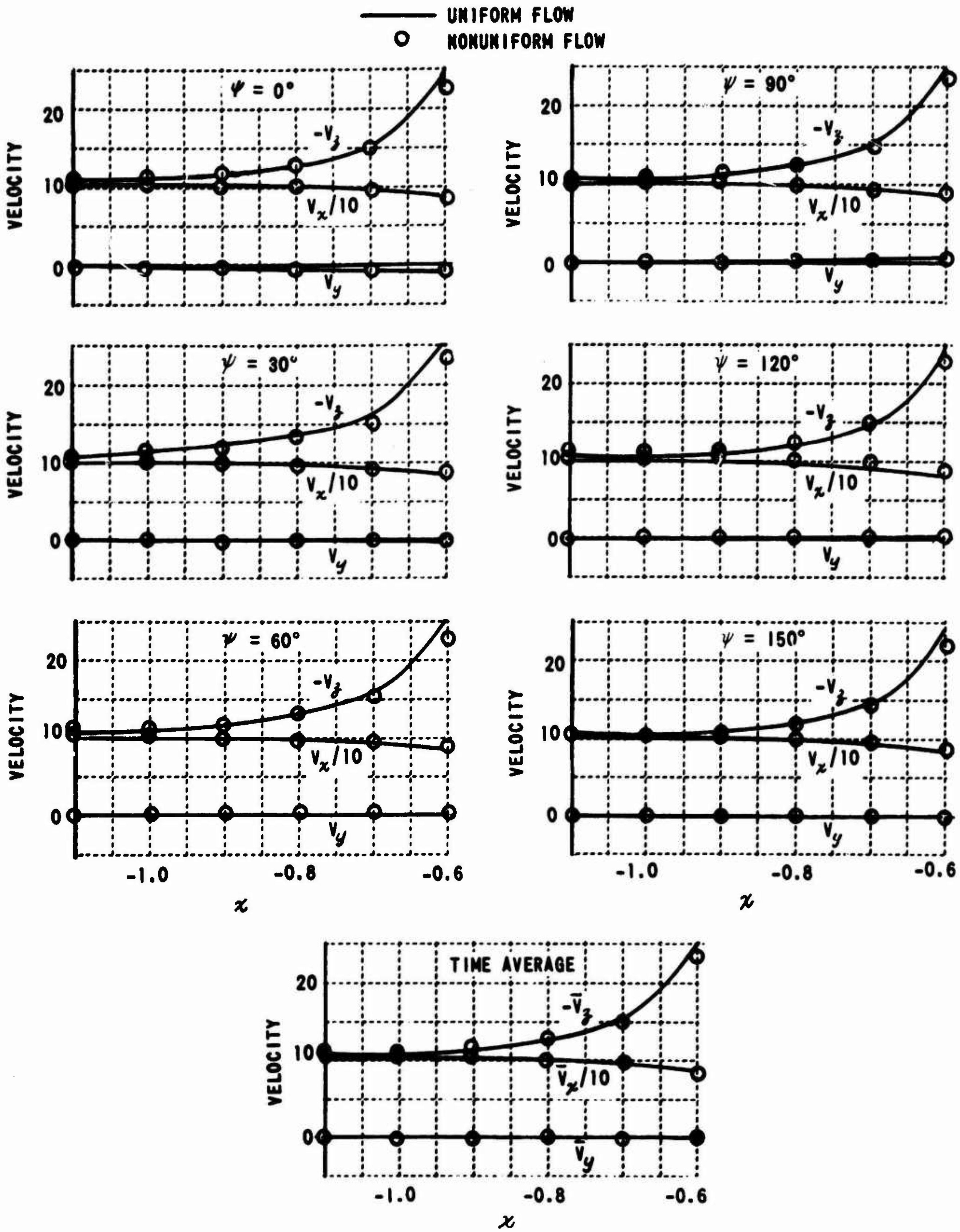

Figure 1 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH x FOR $y = 0$, $z = -0.45$ AND $\mu = 0.220$

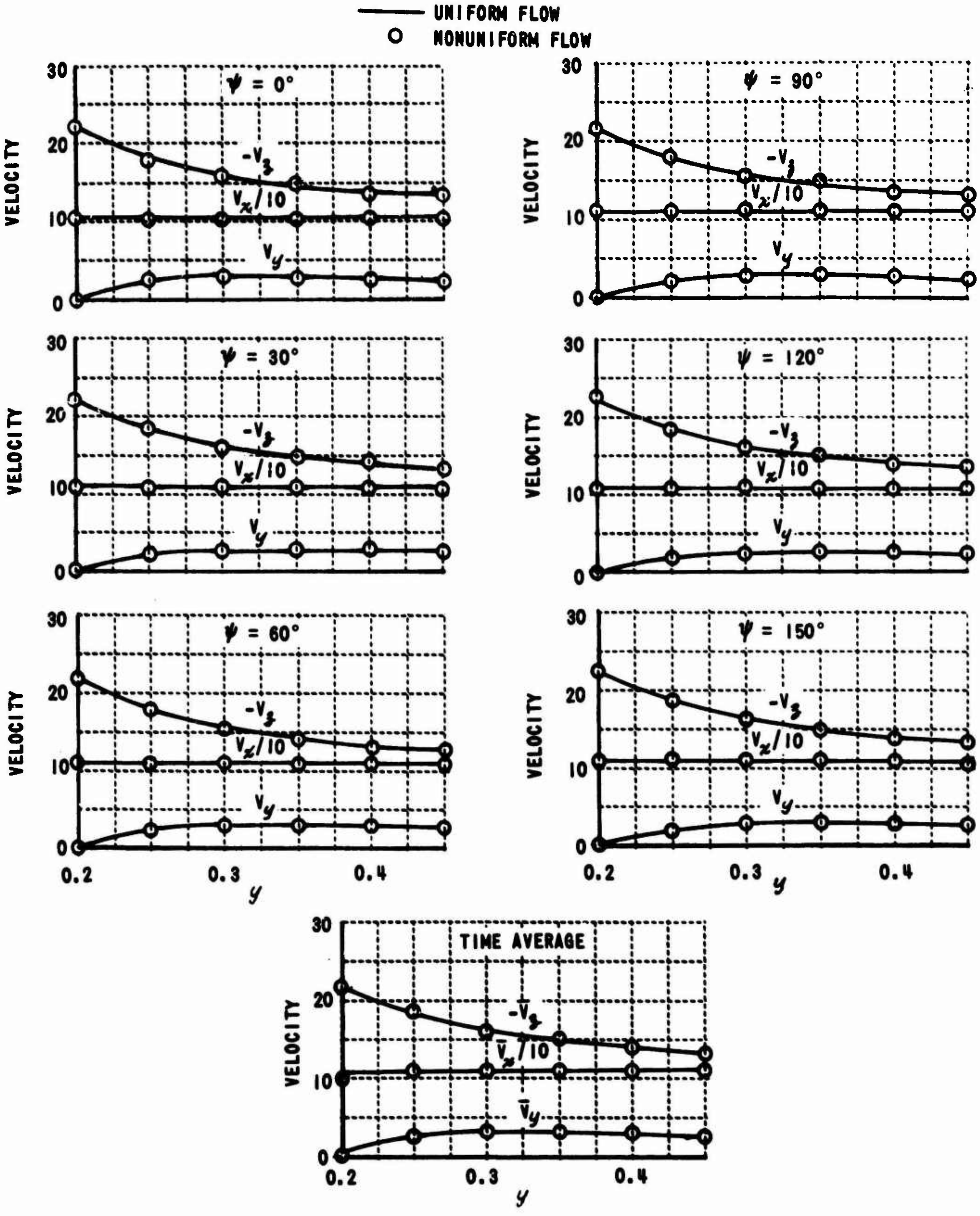

Figure 2 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH y FOR x = -0.25, z = -0.45, μ = 0.220 AND $y > 0$

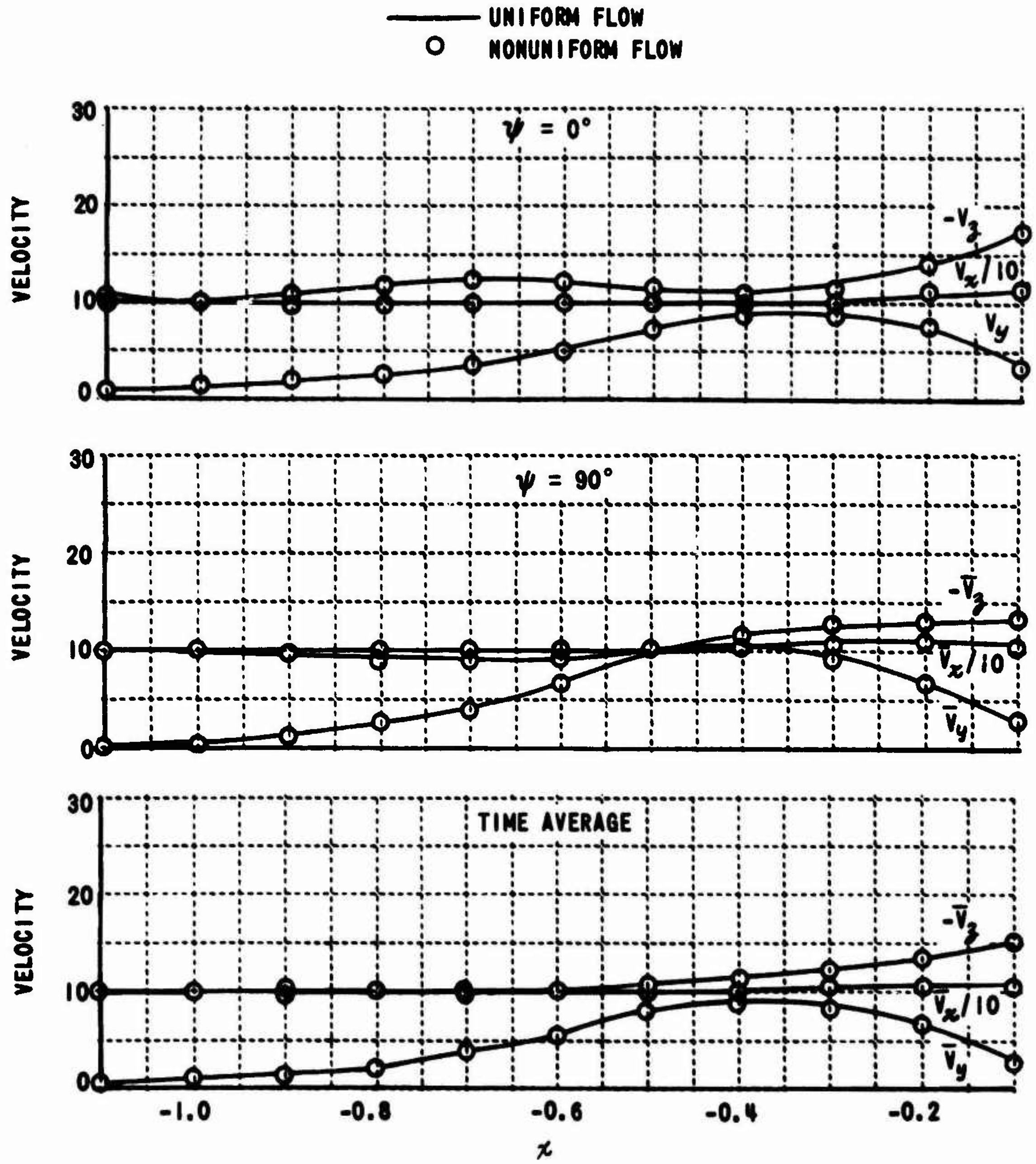

Figure 3 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH x FOR y = 0.3, z = -0.25 AND μ = 0.220

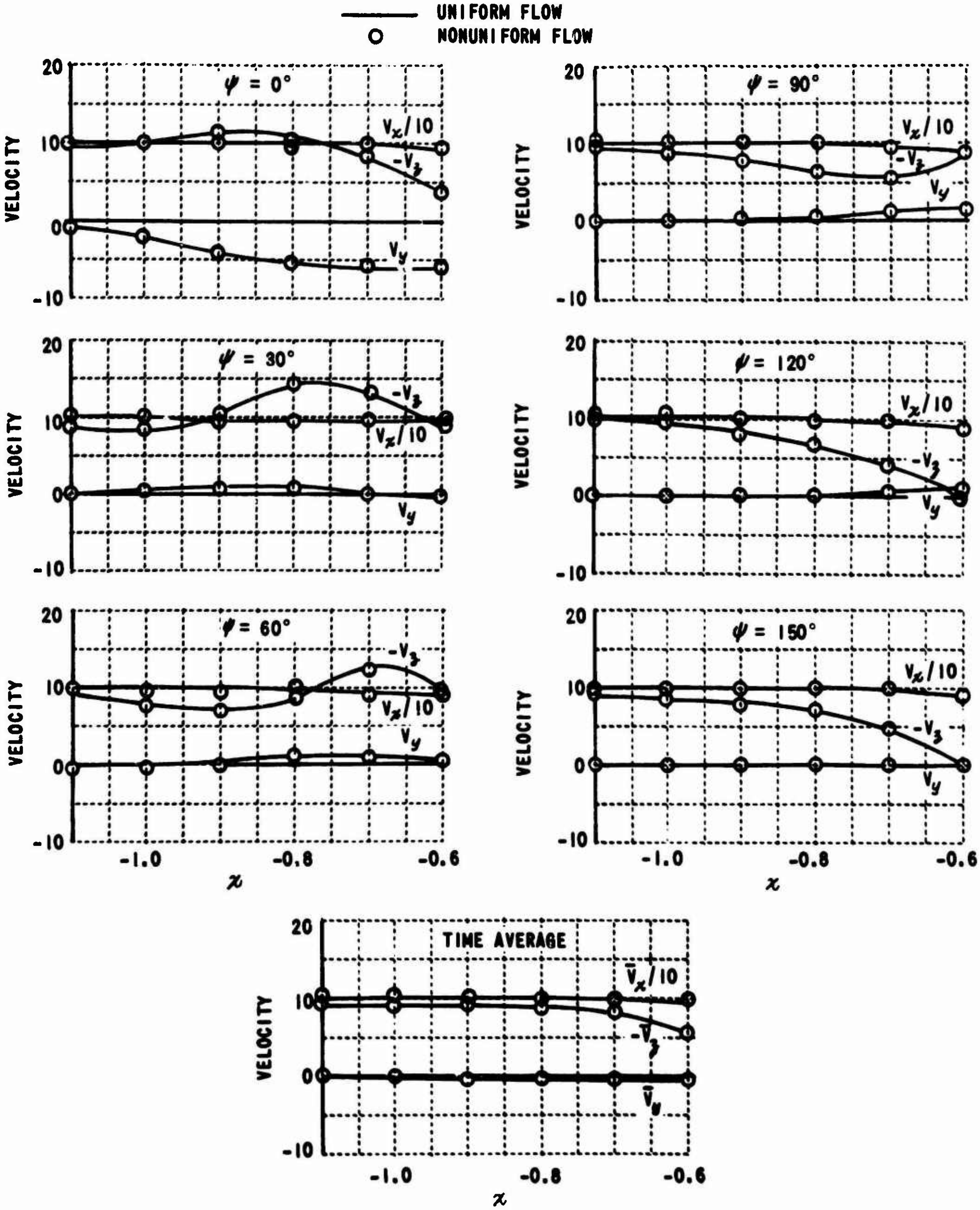

Figure 4 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH x FOR $y = 0$, $z = -0.15$ AND $\mu = 0.220$

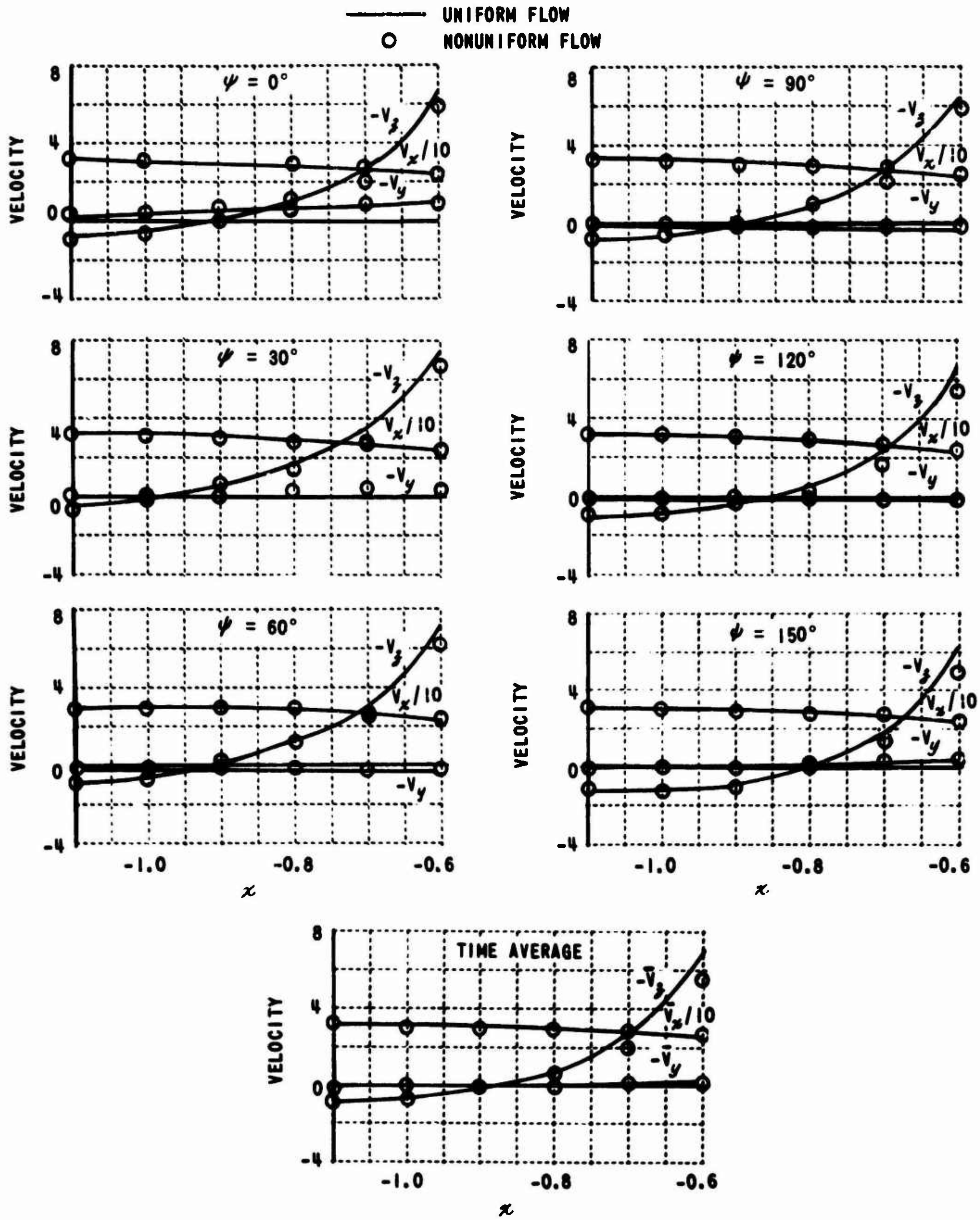

Figure 5 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH x FOR y = 0, z = -0.45 AND μ = 0.0732

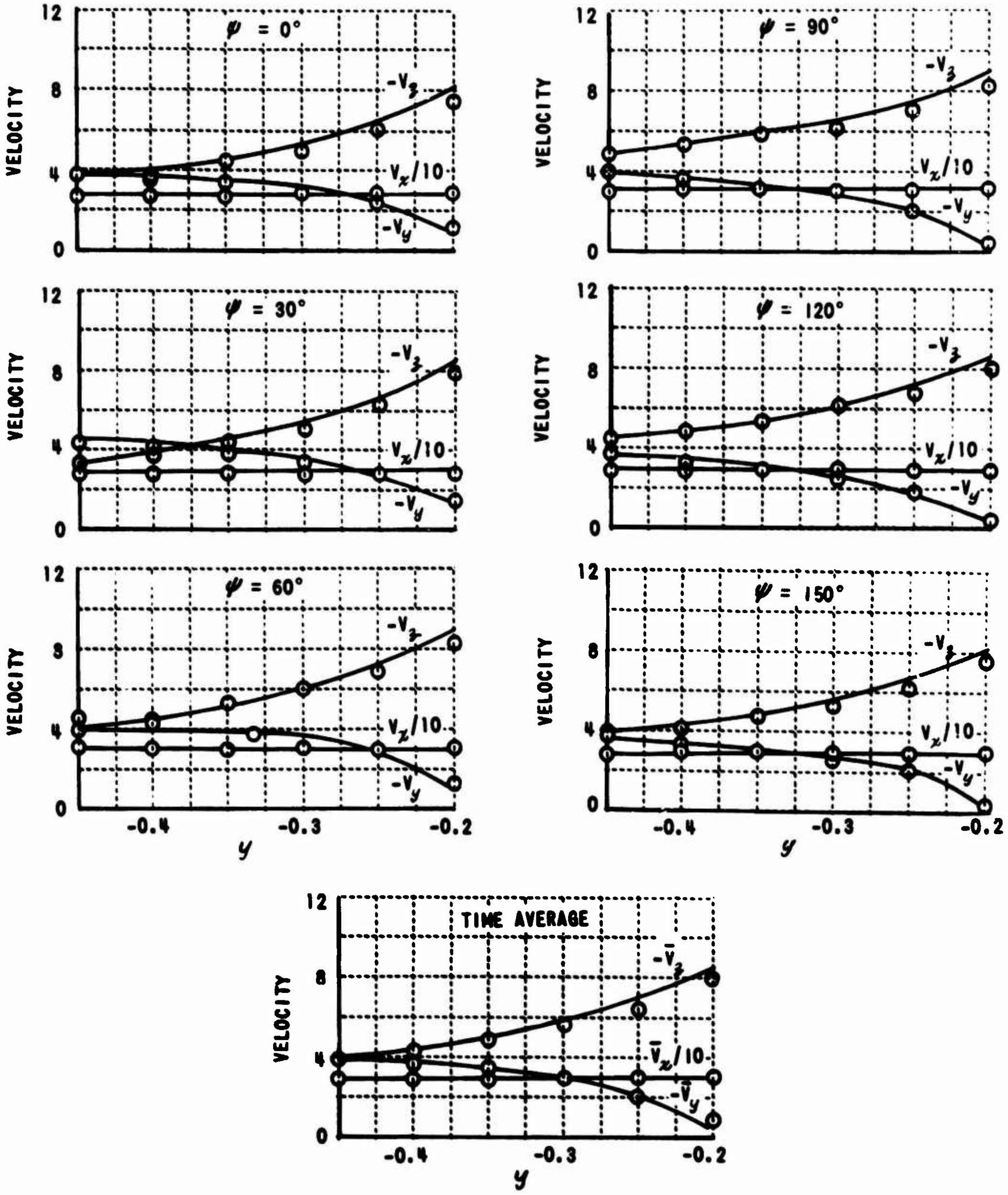

Figure 6 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH y FOR x = -0.25, z = -0.45, μ = 0.0732 AND $y < 0$

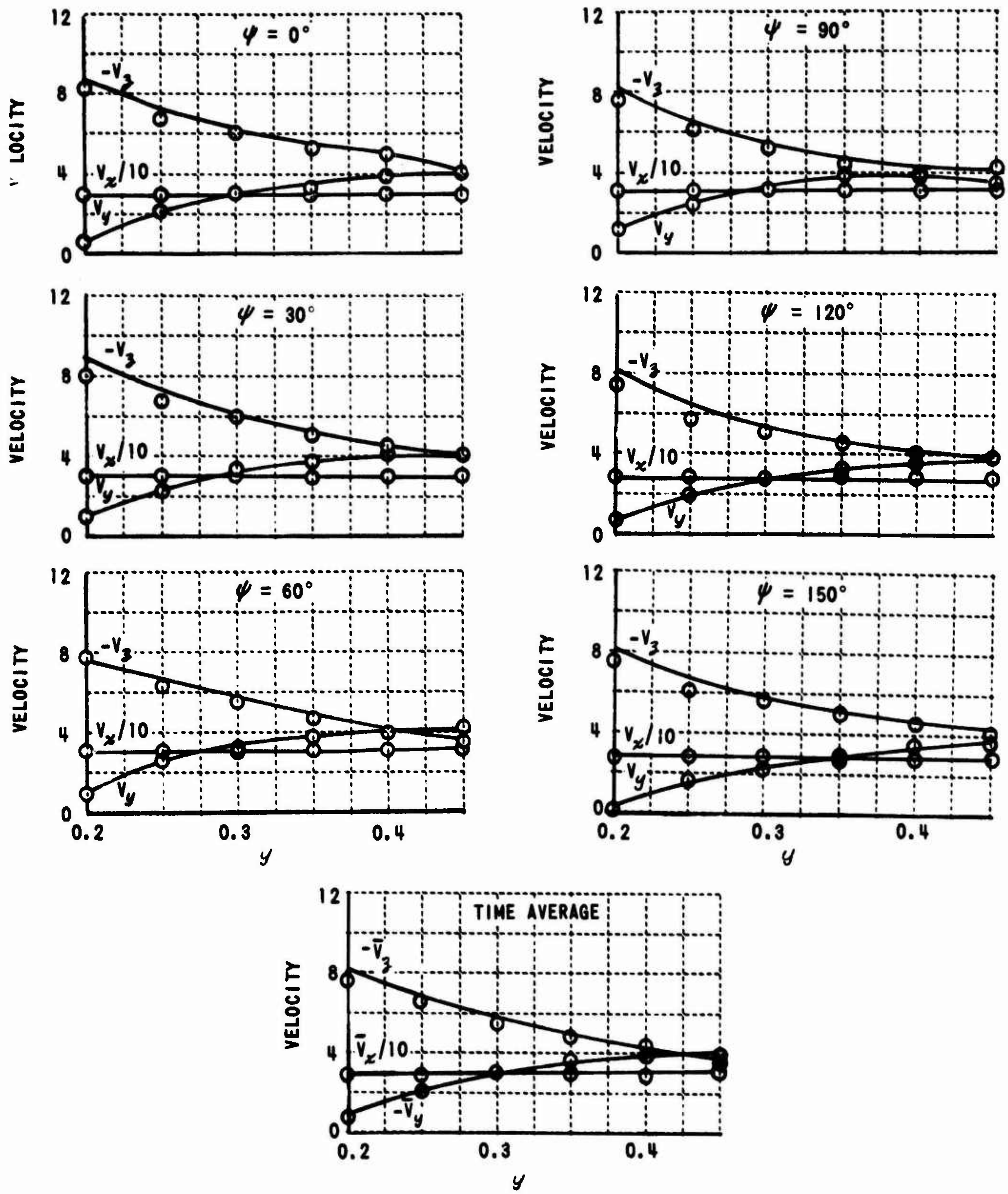

Figure 7 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH y FOR x = -0.25, z = -0.45, μ = 0.0732 AND ψ > 0

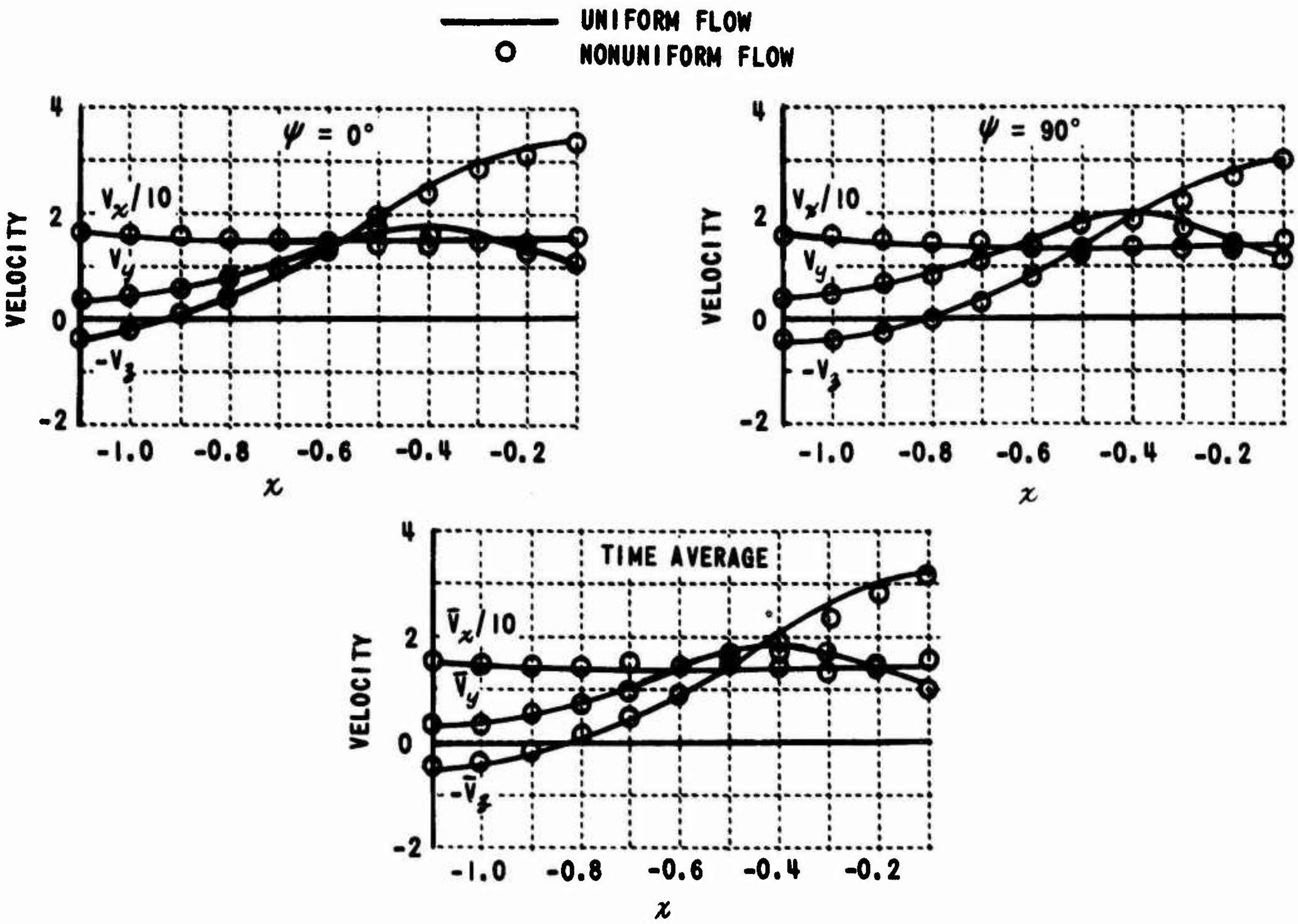

Figure 8 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH x FOR $y = 0.30$, $z = -0.45$ AND $\mu = 0.0732$

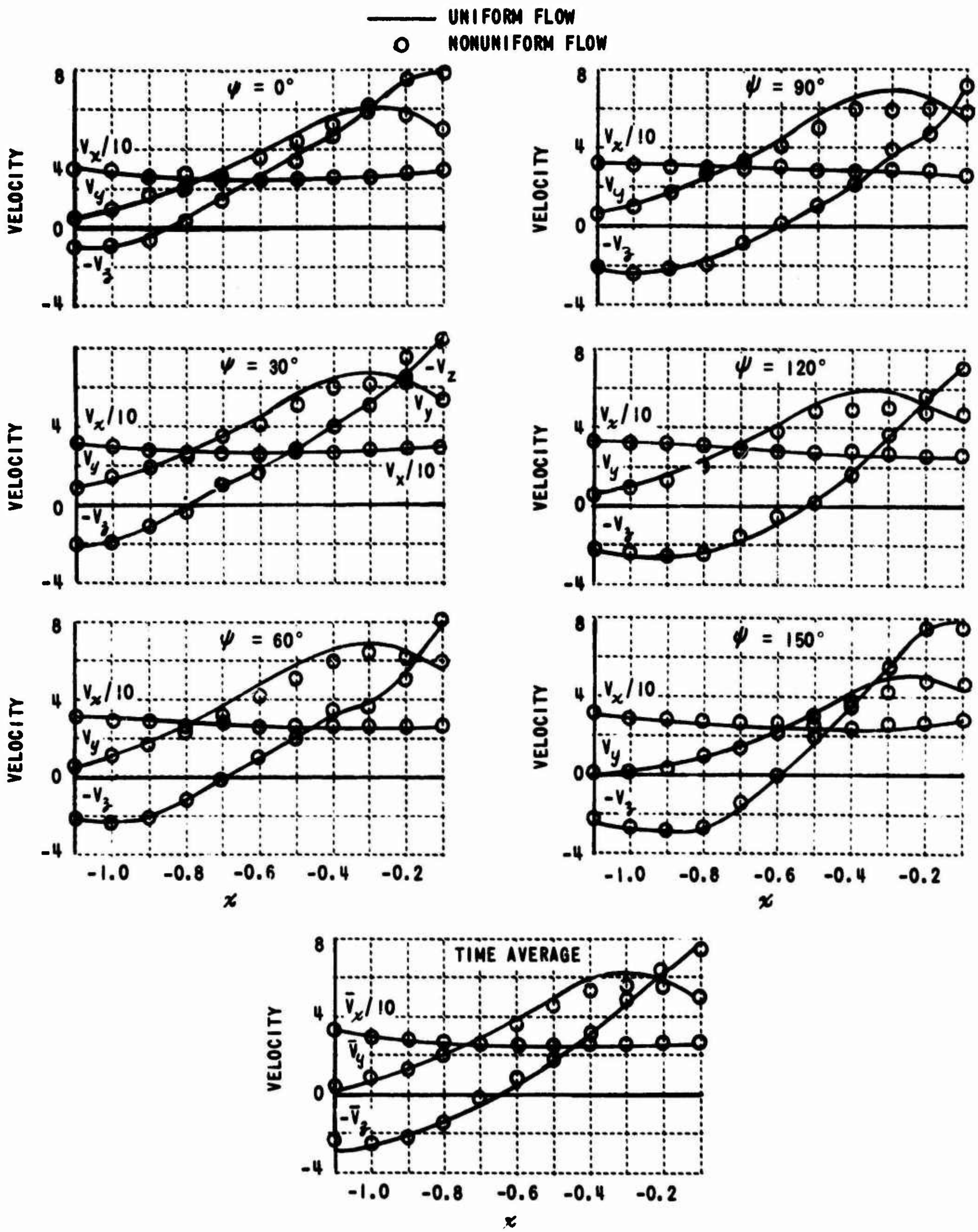

Figure 9 COMPARISON OF THE VARIATION OF THE VELOCITY COMPONENTS WITH x FOR y = 0.30, z = -0.25 AND μ = 0.0732

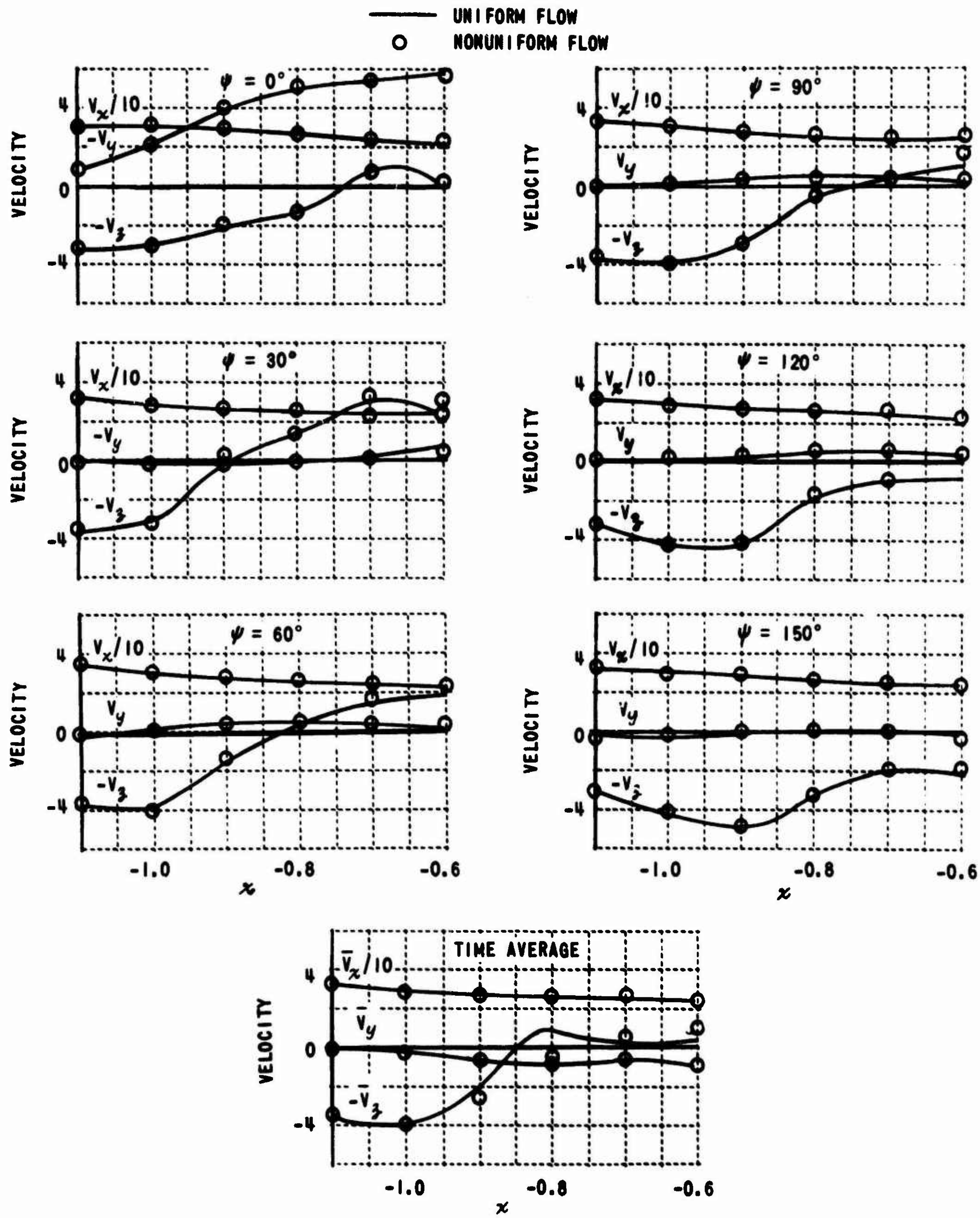

Figure 10 COMPARISON OF THE VARIATION OF VELOCITY COMPONENTS WITH x FOR y = 0, z = -0.15 AND μ = 0.0732

4. CONCLUSIONS AND RECOMMENDATIONS

Based on the comparisons made between the results obtained using a fuselage model for uniform and for nonuniform flow, the following conclusions were drawn.

No differences between the two models were observed in the effects due to the fuselage at distances greater than 0.2R from the fuselage. This was true for both high and low advance ratios.

At the low advance ratio ($\mu = 0.0732$), the nonuniform flow model was found to yield axial velocities which were approximately 14% lower at 0.1R from the fuselage than the uniform flow model. At the high advance ratio ($\mu = 0.220$), the axial velocity was approximately 10% lower at the same locations.

Except for small changes in magnitudes, the character of the flow was the same for both models.

The results of this study show that the effects of the nonuniform flow over the fuselage are significant only in the immediate vicinity of the fuselage (within about 0.1R). Otherwise, the simpler uniform flow model for the fuselage is recommended in view of the reduction in digital computation time.

REFERENCES

1. Crimi, Peter, Theoretical Prediction of the Flow in the Wake of a Helicopter Rotor, Part 1, Cornell Aeronautical Laboratory Report BB-1994-S-1, September 1965.

2. Crimi, Peter, Theoretical Prediction of the Flow in the Wake of a Helicopter Rotor, Part 2, Cornell Aeronautical Laboratory Report BB-1994-S-2, September 1965.

APPENDIX I

OPERATIONAL INFORMATION FOR THE MODIFIED MAIN PROGRAM

This program is written in FORTRAN IV, with the exception of subroutine CLEAR, which is written in MAP. This routine is used to initialize storages to be zero.

INPUTS

CARD 1	NB:	Number of blades, N_B
	NRW:	Number of Revolutions of wake per blade, N_R
	NA:	Number of azimuth stations, N_A
	NPNCH:	Punch option. If zero, no cards are punched at the end of a run. If not zero, all wake point coordinates and core sizes at the final azimuth position are punched on cards.
	NOPT:	If zero, the initial wake configuration is computed. If not zero, initial wake configuration is read in.
	NTAPE:	If not zero, wake point coordinates and velocities are saved on utility Tape 4.
	NPRINT:	If NPRINT = 1, coordinates and velocities for each wake point are printed; if NPRINT = 2, those for every other point are printed; if 3, every third; etc.
	LNCT:	Number of lines desired per page of output.
	NFPT:	Number of fuselage points, N_f
	NXPT:	Number of points off the wake for which velocities are to be calculated.
	NPINT:	Output is produced at intervals of NPINT steps; i. e., if NPINT = 1, the data for each azimuth position is printed.

CARD 2	NFOPT:	Fuselage Option	= 0; indicates $V_{x_\infty} = 1$, $V_{z_\infty} = 0$ ≠ 0; indicates uniform flow option.
	NOUPT:	Output Option	= 0; no print-out ≠ 0; print-out and punch cards.

CARD 3	PSI0:	Initial position of blade 1, ψ_{init} , degrees.
	REV:	Number of revolutions of rotor for which calculations are to be performed, N_{RV}
	XLAM:	λ
	XMU:	μ
	ALPHAT:	α_T (degrees)
	FACTR:	Factor applied to V_{z_∞}, K_f
	RB:	R/b
CARD 4, 5,	GAMB:	Strengths of blade 1; NA of them.
	A1:	Core sizes at Blade 1; (NA of them).
	A:	Initial core sizes; (NRW)(NA)(NB) of them.

Fuselage Data: Four cards for each point; (4)(NFPT) cards in all. These are punched by the Fuselage Program.

Card 1	XBAR	YBAR	ZBAR	SIGX	SIGZ	
	$\bar{x}$	$\bar{y}$	$\bar{z}$	σ_x	σ_z	
Card 2	XI1	XI2	XI3	XI4	ETA1	ETA2
	ξ_1	ξ_2	ξ_3	ξ_4	η_1	η_2

Card 3	ETA3	ETA4	D1	D2	D3	D4
	η_3	η_4	d_1	d_2	d_3	d_4
Card 4	XLE	XME	XNE	XLZ	XMZ	XNZ
	λ_η	μ_η	ν_η	λ_ζ	μ_ζ	ν_ζ

Coordinates of points off the wake at which velocities are to be computed NXPT points in all (up to three sets of coordinates per card):

XIPT	YIPT	ZIPT
x	y	z

Initial Wake Configuration - Read in only if NOPT is not zero.

X: (NRW)(NB)(NA) of them.
Y: (NRW)(NB)(NA) of them.
Z: (NRW)(NB)(NA) of them.

A listing of the program is given on the pages which follow.

```
$IBFTC ZZUWV    LIST,REF
C     CALCULATION OF THE WAKE VORTICITY DISTRIBUTION FOR A HELICOPTER
      COMMON X(340,4),Y(340,4),Z(340,4),U(340,4),V(340,4),W(340,4),
     1       GAMA(340,4),SEG(340,4),GAMB1(100),A1(100),A(340,4),PI,RAD,
     2       VMP,XMCL,XMSL,NB,NRW,NA,NW,NOPT,NTAPE,NPRINT,NDVCH,PSIO,
     3       XMU,XLAM,ALPHAT,PINT,PSIF,XNA,DPSI,NB1,NW1,XNW,XNB,TPNB,
     4       SAT,CAT,C1,C2,PSI,TPI,XI,TI,T2,XJ,NPS,JPS,IPS,IPS1,XXX,YYY,
     5       ZZZ,IST,IND,IFLG,SIG1,SIG2,SIG3,GGG,DEN,XNU1,XNU2,XNU3,IR1,
     6       XMX,XMY,XMZ,SIG4,SIG5,RR,SOI1,SF,SQI,SG,BF,LPS,SFG1,SFG2,
     7       SUM,LNCT,XIPT(400),YIPT(400),ZIPT(400),VX(400),VY(400),
     8       VZ(400),VXF(400),VYF(400),VZF(400),NXPT,NAB,FACTR,RB
      COMMON /FUSE/ XBAR(100),YBAR(100),ZBAR(100),SIGX(100),SIGZ(100),
     1       XI1(100),XI2(100),XI3(100),XI4(100),ETA1(100),ETA2(100),
     2       ETA3(100),ETA4(100),XLE(100),XMF(100),XNE(100),XLZ(100),
     3       XXZ(100),XNZ(100),NFPT,RJ(4),EJ(4),HJ(4),EMJ(4),D1(100),
     4       D2(100),D3(100),D4(100),VXINF,VYINF,VZINF,UF,VF,WF
      DIMENSION GAMB(100)
      EQUIVALENCE(GAMB1,GAMB)
      DIMENSION VXJ(200),VZJ(200)
    1 CALL CLEAR(X,NAB)
      CALL CLEAR(XBAR,WF)
      CALL CLEAR(VXJ(1),VXJ(200))
      CALL CLEAR(VZJ(1),VZJ(200))
      PI = 3.1415926536
      RAD = .0174532925
      TPI = 2.0*PI
      READ 1000,NB,NRW,NA,NPNCH,NOPT,NTAPE,NPRINT,LNCT,NFPT,NXPT,NPINT,
     1          NDVCH
 1000 FORMAT(12I6)
      READ 1000,NFOPT,NOUTP
      CALL DVDCHK(NDVCH)
      IF(NTAPE.LT.0) REWIND 4
      READ 1001,PSIO,REV,XLAM,XMU,ALPHAT,FACTR,RB
 1001 FORMAT(9F8.6)
      READ 1001,(GAMB1(I),I=1,NA)
      READ 1001,(A1(I),I=1,NA)
      NXPT2 = NXPT/2
      NB1 = NB
      NW = NRW*NA
      NW1 = NW+1
      NAB = NB*NA
      ILIM = NA/NB
      ICOUNT = 0
      AVG = .5*FLOAT(NB)/FLOAT(NA)
      XNA = NA
      DPSI = 2.0*PI/XNA
      XNW = NW
      XNB = NB
      SAT = SIN(ALPHAT*RAD)
      CAT = COS(ALPHAT*RAD)
      C1 = XMU*CAT
      C2 = (XMU*SAT+XLAM)
      C3 = XMU*SAT+SQRT(XLAM*XNB/2.0)
      XMCL = C1/XLAM
```

```
      XMSL = XMU*SAT/XLAM
      VXINF = XMCL
      VZINF = -FACTR*(XMSL+SQRT(.5*XNB/XLAM))
      IF (NFOPT.NE.0) GO TO 99
      VXINF = 1.0
      VZINF = 0.0
   99 TMP = RB*DPSI
      TMP1 = SQRT(TMP*(TMP+2.0))
      FRB = (TMP-TMP1+ALOG(1.0+TMP+TMP1))/DPSI
      READ 1001,((A(I,J),I=1,NW),J=1,NB1)
      IF(NFPT.EQ.0) GO TO 100
      READ 1003,(XBAR(I),YBAR(I),ZBAR(I),SIGX(I),SIGZ(I),BLNK,XI1(I),
     1           XI2(I),XI3(I),XI4(I),ETA1(I),ETA2(I),ETA3(I),ETA4(I),
     2           D1(I),D2(I),D3(I),D4(I),XLE(I),XME(I),XNE(I),XLZ(I),
     3           XXZ(I),XNZ(I),I=1,NFPT)
 1003 FORMAT(6E12.5)
  100 IF(NXPT.EQ.0) GO TO 103
      READ 1001,(XIPT(I),YIPT(I),ZIPT(I),I=1,NXPT)
  101 DO 102  I=1,NXPT
      CALL FUSLGE(XIPT(I),YIPT(I),ZIPT(I),VXF(I),VYF(I),VZF(I))
  102 CONTINUE
  103 PSIF = PSI0+360.0*REV
      CALL IDOUT
      NCT = 0
      TPNB = 2.0*PI/XNB
      PSI = PSI0*RAD
      PSI0 = PSI
      PSIF = PSIF*RAD+0.05
      IF(NOPT.EQ.0) GO TO 3
    2 READ 1003,((X(I,J),I=1,NW1),J=1,NB)
      READ 1003,((Y(I,J),I=1,NW1),J=1,NB)
      READ 1003,((Z(I,J),I=1,NW1),J=1,NB)
      GO TO 7
    3 DO 6  I=1,NW1
      XI = FLOAT(I-1)*DPSI
      T1 = XI*C1
      T3 = XI*C3
    4 DO 5  J=1,NB
      XJ = FLOAT(J-1)*TPNB
      X(I,J) = COS(PSI0+XJ-XI)+T1
      Y(I,J) = SIN(PSI0+XJ-XI)
      Z(I,J) = -T3
    5 CONTINUE
    6 CONTINUE
    7 NPS = AMOD(PSI0,2.0*PI)/DPSI+1.5
      IF (NPS.GT.NA) NPS = 1
      DO 9  J=1,NB
      JPS = MOD(NPS+(NA*(J-1))/NB+NAB,NA)
      IF (JPS.EQ.0) JPS = NA
      IPS1 = JPS
      DO 8  I=1,NW
      IPS = IPS1
      IPS1 = IPS-1
      IF(IPS1.EQ.0) IPS1 = NA
```

```
      GAMA(I,J) = (GAMB(IPS)+GAMB(IPS1))/2.0
    8 CONTINUE
    9 CONTINUE
   10 DO 12  J=1,NB1
      DO 11  I=1,NW
      SEG(I,J) = SQRT((X(I,J)-X(I+1,J))**2+(Y(I,J)-Y(I+1,J))**2+(Z(I,J)-
     1           Z(I+1,J))**2)
   11 CONTINUE
   12 CONTINUE
   13 DO 29  I=1,NW
      DO 28 J=1,NB1
      XXX = X(I,J)
      YYY = Y(I,J)
      ZZZ = Z(I,J)
      U(I,J) = 0.0
      V(I,J) = 0.0
      W(I,J) = 0.0
   14 DO 25  L=1,NB1
      IST = 1
      IND = NW
      IFLG = 1
      IF (L.NE.J) GO TO 16
      IND = I-2
      IFLG = 2
      IF (IND.GT.0) GO TO 16
   15 IST = I+1
      IND = NW
      IFLG = 1
      IF (IST.GT.NW) GO TO 18
   16 SIG2 = SQRT((XXX-X(IST,L))**2+(YYY-Y(IST,L))**2+(ZZZ-Z(IST,L))**2)
      DO 17 IR=IST,IND
      SIG1 = SIG2
      SIG2 = SQRT((XXX-X(IR+1,L))**2+(YYY-Y(IR+1,L))**2+(ZZZ-Z(IR+1,L))
     1       **2)
      SEGSQ = SEG(IR,L)**2
      HM1 = SIG1**2+SIG2**2
      IF(HM1.GT.SEGSQ)GO TO 160
      HM2 = .25*((SIG1+SIG2)**2-SEGSQ)*(SEGSQ-(SIG1-SIG2)**2)/SEGSQ
      IF(HM2.GT.A(IR,L)**2)GO TO 160
      GGG = GAMA(IR,L)/SEG(IR,L)
      GO TO 161
  160 GGG = GAMA(IR,L)*(SIG1+SIG2)/(SIG1*SIG2*((SIG1+SIG2)**2-SEGSQ))
  161 XNU1 = (YYY-Y(IR+1,L))*(Z(IR,L)-Z(IR+1,L))-(ZZZ-Z(IR+1,L))*
     1         (Y(IR,L)-Y(IR+1,L))
      XNU2 = (ZZZ-Z(IR+1,L))*(X(IR,L)-X(IR+1,L))-(XXX-X(IR+1,L))*
     1       (Z(IR,L)-Z(IR+1,L))
      XNU3 = (XXX-X(IR+1,L))*(Y(IR,L)-Y(IR+1,L))-(YYY-Y(IR+1,L))*
     1       (X(IR,L)-X(IR+1,L))
      U(I,J) = U(I,J)+XNU1*GGG
      V(I,J) = V(I,J)+XNU2*GGG
      W(I,J) = W(I,J)+XNU3*GGG
   17 CONTINUE
      GO TO (18,15),IFLG
   18 IF (L.NE.J) GO TO 25
```

```
      IR1 = I-1
      IF (I.EQ.1) IR1 = 1
      XMX = (Y(IR1,L)-Y(IR1+1,L))*(Z(IR1+1,L)-Z(IR1+2,L))-(Y(IR1+1,L)-
     1      Y(IR1+2,L))*(Z(IR1,L)-Z(IR1+1,L))
      XMY = (Z(IR1,L)-Z(IR1+1,L))*(X(IR1+1,L)-X(IR1+2,L))-(Z(IR1+1,L)-
     1      Z(IR1+2,L))*(X(IR1,L)-X(IR1+1,L))
      XMZ = (X(IR1,L)-X(IR1+1,L))*(Y(IR1+1,L)-Y(IR1+2,L))-(X(IR1+1,L)-
     1      X(IR1+2,L))*(Y(IR1,L)-Y(IR1+1,L))
      SIG4 = SFG(IR1+1,L)
      SIG3 = SFG(IR1,L)
      SIG5 = SQRT((X(IR1+2,L)-X(IR1,L))**2+(Y(IR1+2,L)-Y(IR1,L))**2+
     1       (Z(IR1+2,L)-Z(IR1,L))**2)
      DEN = (SIG3+SIG4-SIG5)*(SIG3+SIG4+SIG5)*(SIG4+SIG5-SIG3)*(SIG3+
     1      SIG5-SIG4)
      IF (DEN.EQ.0.0) GO TO 25
      IF(DEN.LT.0.0)WRITE(6,1002)I,J,SIG3,SIG4,SIG5
 1002 FORMAT(2X43HDENOMINATOR NEGATIVE FOR R COMPUTATION   I =I3,3X3HJ =
     1       I3,3X6HSIG3 =E16.8,3X6HSIG4 =E16.8,3X6HSIG5 =E16.8 )
      RR = SIG3*SIG4*SIG5/SQRT(ABS(DEN))
      SQI1 = SQRT((2.0*RR-SIG3)*(2.0*RR+SIG3))
      IF (SIG3**2.LE.SIG4**2+SIG5**2) GO TO 19
      SF = (2.0*RR+SQI1)/SIG3
      GO TO 20
   19 SF = (2.0*RR-SQI1)/SIG3
      IF(SF.EQ.0.0) SF = 1.0E-20
   20 SQI = SQRT((2.0*RR-SIG4)*(2.0*RR+SIG4))
      IF (SIG4**2.LE.SIG3**2+SIG5**2) GO TO 21
      SG = (2.0*RR+SQI)/SIG4
      GO TO 22
   21 SG = (2.0*RR-SQI)/SIG4
      IF(SG.EQ.0.0) SG = 1.0E-20
   22 IF (I.EQ.1) GO TO 23
      BF = (GAMA(IR1,J)*(ALOG(8.0*SF/A(IR1,J))+.25)+GAMA(I,J)*(ALOG(8.0*
     1     SG/A(I,J))+.25))/(4.0*RR*SQRT(XMX**2+XMY**2+XMZ**2))
      GO TO 24
   23 BF = (GAMA(I,J)*(ALOG(8.0*SF/A(I,J))+.25))/(4.0*RR*SQRT(XMX**2+
     1     XMY**2+XMZ**2))
   24 U(I,J) = U(I,J)+XMX*BF
      V(I,J) = V(I,J)+XMY*BF
      W(I,J) = W(I,J)+XMZ*BF
   25 CONTINUE
      SIG1 = SQRT(XXX**2+YYY**2+ZZZ**2)
   26 DO 27  L=1,NB
      LPS = MOD(NPS+(NA*(L-1))/NB+NAB,NA)
      IF(LPS.EQ.0) LPS = NA
      IF(I.EQ.1.AND.L.EQ.J)GO TO 260
      PSIBK = FLOAT(LPS-1)*DPSI
      SINPSI = SIN(PSIBK)
      COSPSI = COS(PSIBK)
      RMH2 = (XXX-COSPSI)**2+(YYY-SINPSI)**2+ZZZ**2
      IF (RMH2+SIG1**2.GT.1.0) GO TO 258
      RMH = SQRT(RMH2)
      H2 = .25*((SIG1+RMH)**2-1.0)*(1.0-(SIG1-RMH)**2)
      IF (H2*RB**2.GT.1.0) GO TO 258
```

```
      HH = SQRT(H2)
      XHT = XXX*(COSPSI**2+SINPSI**2/(HH*RB))-YYY*SINPSI*COSPSI*(1.0/
     1        (HH*RB)-1.0)
      YHT = YYY*(SINPSI**2+COSPSI**2/(HH*RB))-XXX*SINPSI*COSPSI*(1.0/
     1        (HH*RB)-1.0)
      ZHT = ZZZ/(HH*RB)
      XNU1 = -YHT*Z(1,L)+ZHT*Y(1,L)
      XNU2 = -ZHT*X(1,L)+XHT*Z(1,L)
      XNU3 =-XHT*Y(1,L)+YHT*X(1,L)
      SIG2 = SQRT((XHT-X(1,L))**2+(YHT-Y(1,L))**2+(ZHT-Z(1,L))**2)
      GO TO 259
  258 XNU1 = -YYY*Z(1,L)+ZZZ*Y(1,L)
      XNU2 = -ZZZ*X(1,L)+XXX*Z(1,L)
      XNU3 = -XXX*Y(1,L)+YYY*X(1,L)
      SIG2 = SQRT((XXX-X(1,L))**2+(YYY-Y(1,L))**2+(ZZZ-Z(1,L))**2)
  259 GGG = GAMB(LPS)*(SIG1+SIG2)/(SIG1*SIG2*((SIG1+SIG2)**2-1.0))
      U(I,J) = U(I,J)+XNU1*GGG
      V(I,J) = V(I,J)+XNU2*GGG
      W(I,J) = W(I,J)+XNU3*GGG
      GO TO 27
  260 W(I,J) = W(I,J)-GAMB(LPS)*FRB
   27 CONTINUE
      CALL FUSLGE(X(I,J),Y(I,J),Z(I,J),UF,VF,WF)
      U(I,J) = U(I,J)+XMCL+UF
      V(I,J) = V(I,J)+VF
      W(I,J) = W(I,J)-XMSL+WF
   28 CONTINUE
   29 CONTINUE
      IF(NTAPE.EQ.0) GO TO 30
      WRITE(4)PSI,XMU,XLAM,ALPHAT,NB,NRW,NA,NW
      WRITE(4) ((X(I,J),Y(I,J),Z(I,J),U(I,J),V(I,J),W(I,J),GAMA(I,J),
     1          A(I,J),I=1,NW1),J=1,NB1)
   30 IF(NCT.NE.0)GO TO 31
      IF(NXPT.EQ.0) GO TO 302
      CALL VLCTY
      IF (NOUTP.EQ.0) GO TO 300
      IF (ICOUNT.GE.ILIM) GO TO 302
  300 DO 301  I=1,NXPT2
      J = 2*I-1
      VXJ(I) = VXJ(I)+AVG*(VX(J)+VX(J+1))
      VZJ(I) = VZJ(I)+AVG*(VZ(J)+VZ(J+1))
  301 CONTINUE
      ICOUNT = ICOUNT+1
  302 CALL OUTPUT
   31 NCT = NCT+1
      IF(NCT.GE.NPINT)NCT = 0
      PSI = PSI+DPSI
      NPS = NPS+1
      IF (NPS.GT.NA) NPS = 1
      IF(PSI.LE.PSIF) GO TO 32
      IF(NTAPE.NE.0) END FILE 4
      IF(NPNCH.EQ.0) GO TO 320
      PUNCH 1004
 1004 FORMAT(74HZZ    HELICOPTER WAKE VORTICITY CALCULATIONS  -  HARVEY
```

```
1SELIR          Z7 )
      PUNCH 1001,((A(I,J),I=1,NW),J=1,NB)
      PUNCH 1003,((X(I,J),I=1,NW1),J=1,NB)
      PUNCH 1003,((Y(I,J),I=1,NW1),J=1,NB)
      PUNCH 1003,((Z(I,J),I=1,NW1),J=1,NB)
  320 IF (NOUTP.EQ.0) GO TO 1
      PUNCH 1005
 1005 FORMAT(80H*****************************************************************
     1****************ZZZZZZZZ)
      PUNCH 1006,(VXJ(I),VZJ(I),I=1,NXPT)
 1006 FORMAT(6F12.5)
      WRITE(6,1007) ILIM,(VXJ(I),VZJ(I),I=1,NXPT2)
 1007 FORMAT(1H1,39X39HVELOCITY AVERAGES FOR OTHER POINTS OVER I3,
     1        9H AZIMUTHS//54X2HHVX,20X2HVZ/(F59.5,F22.5))
      GO TO 1
   32 DO 35  J=1,NB1
      TX1 = X(1,J)
      TY1 = Y(1,J)
      TZ1 = Z(1,J)
      DO 33 I=2,NW1
      TX2 = TX1
      TY2 = TY1
      TZ2 = TZ1
      TX1 = X(I,J)
      TY1 = Y(I,J)
      TZ1 = Z(I,J)
      X(I,J) = TX2+XLAM*U(I-1,J)*DPSI
      Y(I,J) = TY2+XLAM*V(I-1,J)*DPSI
      Z(I,J) = TZ2+XLAM*W(I-1,J)*DPSI
   33 CONTINUE
      XJ = FLOAT(J-1)*TPNB
      X(1,J) = COS(PSI+XJ)
      Y(1,J) = SIN(PSI+XJ)
      Z(1,J) = 0.0
   35 CONTINUE
      DO 38  J=1,NB1
      JPS = MOD(NPS+(NA*(J-1))/NB+NAB,NA)
      IF(JPS.EQ.0) JPS = NA
      JPS1 = JPS-1
      IF(JPS1.EQ.0)JPS1 = NA
      SEG1 = SEG(1,J)
      GAM1 = GAMA(1,J)
      TA1 = A(1,J)
   36 DO 37  I=2,NW
      GAM2 = GAM1
      GAM1 = GAMA(I,J)
      GAMA(I,J) = GAM2
      SEG2 = SEG1
      SEG1 = SEG(I,J)
      SEG(I,J) = SQRT((X(I,J)-X(I+1,J))**2+(Y(I,J)-Y(I+1,J))**2+(Z(I,J)-
     1           Z(I+1,J))**2)
      TA2 = TA1
      TA1 = A(I,J)
      A(I,J) = TA2*SQRT(SEG2/SEG(I,J))
```

```
   37 CONTINUE
      SEG(1,J) = SQRT((X(1,J)-X(2,J))**2+(Y(1,J)-Y(2,J))**2+(Z(1,J)-
     1          Z(2,J))**2)
      A(1,J) = A1(JPS)
      GAMA(1,J) = (GAMB(JPS)+GAMB(JPS1))/2.0
   38 CONTINUE
      GO TO 13
      END
```

```
$IBFTC ZZIOT   LIST,REF
C       WAKE VORTICITY CALCULATION PROGRAM - SUBROUTINE IOOUT
        SUBROUTINE IOOUT
        COMMON X(340,4),Y(340,4),Z(340,4),U(340,4),V(340,4),W(340,4),
      1        GAMA(340,4),SFG(340,4),GAMB1(100),A1(100),A(340,4),PI,RAD,
      2         VMP,XMCL,XMSL,NB,NRW,NA,NW,NOPT,NTAPE,NPRINT,NDVCH,PSIO,
      3         XMU,XLAM,ALPHAT,PINT,PSIF,XNA,DPSI,NB1,NW1,XNW,XNB,TPNB,
      4         SAT,CAT,C1,C2,PSI,TPI,XI,T1,T2,XJ,NPS,JPS,IPS,IPS1,XXX,YYY,
      5         ZZZ,IST,IND,IFLG,SIG1,SIG2,SIG3,GGG,DEN,XNU1,XNU2,XNU3,IR1,
      6         XMX,XMY,XMZ,SIG4,SIG5,RR,SQI1,SF,SQI,SG,BF,LPS,SEG1,SEG2,
      7         SUM,LNCT,XIPT(400),YIPT(400),ZIPT(400),VX(400),VY(400),
      8         VZ(400),VXF(400),VYF(400),VZF(400),NXPT,NAB,FACTR,RB
        COMMON /FUSE/ XBAR[illegible],YBAR(100),ZBAR(100),SIGX(100),SIGZ(100),
      1         XI1(100),XI[illegible]),XI3(100),XI4(100),ETA1(100),ETA2(100),
      2         ETA3(100),ETA4(100),XLF(100),XMF(100),XNF(100),XLZ(100),
      3         XXZ(100),XNZ(100),NFPT,RJ(4),EJ(4),HJ(4),EMJ(4),D1(100),
      4         D2(100),D3(100),D4(100),VXINF,VYINF,VZINF,UF,VF,WF
    1 WRITE(6,1000)NB,NRW,NA,NFPT,PSIO,PSIF,XLAM,XMU,ALPHAT,RB,FACTR
 1000 FORMAT(1H1,49X33HHELICOPTER WAKE VORTICITY PROGRAM //45X31HNUMBER
     1OF BLADES                   =I11 /45X31HNUMBER OF REVOLUTIONS OF WAKE =
     2 I11 /45X31HNUMBER OF AZIMUTH STATIONS       =I11/45X31HNUMBER OF FUS
     XELAGE POINTS       =I11/45X23HPSI (INITIAL)             =
     3          F11.3,8H DEGREES /45X23HPSI (FINAL)              =F11.3,
     4 8H DEGREES /45X23HLAMBDA                   =F12.5 /45X23HMU
     5          =F12.5 /45X23HALPHAT                  =F11.3,8H DEGREES /
     6          45X23HR/B                     =F11.3/45X23HVZ INFINITY FACTO
     7R     =F11.3)
    2 DPS = 360.0/FLOAT(NA)
      PS = 0.0
      WRITE(6,1001)
 1001 FORMAT(//30X3HPSI,15X22H  STRENGTH OF BLADE 1 ,14X20HCORE SIZE AT
     1BLADE 1 )
    3 DO 4  I=1,NA
      WRITE(6,1002)PS,GAMB1(I),A1(I)
 1002 FORMAT(F35.3,E30.5,E35.5)
      PS = PS+DPS
    4 CONTINUE
      RETURN
      END
```

```
$IBFTC ZZOUTP  LIST,REF
C     WAKE VORTICITY CALCULATION PROGRAM - SUBROUTINE OUTPUT
      SUBROUTINE OUTPUT
      COMMON X(340,4),Y(340,4),Z(340,4),U(340,4),V(340,4),W(340,4),
     1       GAMA(340,4),SEG(340,4),GAMB1(100),A1(100),A(340,4),PI,RAD,
     2        VMP,XMCL,XMSL,NB,NRW,NA,NW,NOPT,NTAPE,NPRINT,NDVCH,PSIO,
     3        XMU,XLAM,ALPHAT,PINT,PSIF,XNA,DPSI,NB1,NW1,XNW,XNB,TPNB,
     4        SAT,CAT,C1,C2,PSI,TPI,XI,T1,T2,XJ,NPS,JPS,IPS,IPS1,XXX,YYY,
     5        ZZZ,IST,IND,IFLG,SIG1,SIG2,SIG3,GGG,DEN,XNU1,XNU2,XNU3,IR1,
     6        XMX,XMY,XMZ,SIG4,SIG5,RR,SQI1,SF,SQI,SG,BF,LPS,SEG1,SEG2,
     7        SUM,LNCT,XIPT(400),YIPT(400),ZIPT(400),VX(400),VY(400),
     8        VZ(400),VXF(400),VYF(400),VZF(400),NXPT,NAB,FACTR,RB
    1 ILINE = 0
      PSID = PSI/RAD
      DPSID = DPSI/RAD
    2 DO 5 J=1,NB1
      IF(ILINE.EQ.0) WRITE(6,1000)NB,NA,NRW,XLAM,XMU,ALPHAT,DPSID,PSID
 1000 FORMAT(1H1,46X38HHELICOPTER WAKE VORTICITY DISTRIBUTION //13X
     1          15HNO. OF BLADES = I2,23X25HNO. OF AZIMUTH STATIONS = I3,
     2        21X21HNO. OF REV. OF WAKE = I2 /9X8HLAMBDA =E12.5,15X4HMU =
     3      E12.5,12X9HALPHA T = F7.3,5H DEG. 11X11HDELTA PSI = F7.3,
     4     5H DEG. //55X5HPSI = F8.3,8H DEGREES )
    3 WRITE(6,1002)J
 1002 FORMAT( /59X12HBLADE NUMBER I2 / )
    4 WRITE(6,1003)(I,X(I,J),Y(I,J),Z(I,J),U(I,J),V(I,J),W(I,J),
     1                 GAMA(I,J),A(I,J),I=1,NW1,NPRINT)
 1003 FORMAT( 4X5HSTAT.,10X1HX,14X1HY,14X1HZ,14X2HVX,13X2HVY,13X2HVZ,
     1          10X8HSTRENGTH, 6X9HCORE SIZE /(I8,E18.5,7E15.5) )
      ILINE = ILINE + NW1/MAX0(NPRINT,1)+3
      IF (ILINE.GE.LNCT) ILINE = 0
    5 CONTINUE
      IF(NXPT.EQ.0) GO TO 6
      IF(ILINE.EQ.0)WRITE(6,1000)NB,NA,NRW,XLAM,XMU,ALPHAT,DPSID,PSID
      WRITE(6,1004)(XIPT(I),YIPT(I),ZIPT(I),VX(I),VY(I),VZ(I),I=1,NXPT)
 1004 FORMAT( /53X26HVELOCITIES AT OTHER POINTS //19X1HX,14X1HY,14X1HZ,
     1          14X2HVX,13X2HVY,13X2HVZ /(E26.5,5E15.5 ) )
    6 RETURN
      END
```

```
$IBFTC ZZFSLG  LIST,REF
C     WAKE VORTICITY CALCULATION PROGRAM - SUBROUTINE FUSLGE
      SUBROUTINE FUSLGE(XD,YD,ZD,UFD,VFD,WFD)
      COMMON /FUSE/ XBAR(100),YBAR(100),ZBAR(100),SIGX(100),SIGZ(100),
     1        XI1(100),XI2(100),XI3(100),XI4(100),ETA1(100),ETA2(100),
     2        ETA3(100),ETA4(100),XLE(100),XME(100),XNE(100),XLZ(100),
     3        XMZ(100),XNZ(100),NFPT,RJ(4),EJ(4),HJ(4),EMJ(4),D1(100),
     4        D2(100),D3(100),D4(100),VXINF,VYINF,VZINF,UF,VF,WF
      DIMENSION XIK(100,4),ETAK(100,4),DDJ(100,4)
      EQUIVALENCE (XIK,XI1),(ETAK,ETA1),(DDJ,D1)
    1 SUMU = 0.0
      SUMV = 0.0
      SUMW = 0.0
      IF(NFPT.EQ.0) GO TO 13
    2 DO 12  J=1,NFPT
      NFLG = 1
      XLX = XME(J)*XNZ(J)-XMZ(J)*XNE(J)
      XMX = XNE(J)*XLZ(J)-XNZ(J)*XLE(J)
      XNX = XLE(J)*XMZ(J)-XLZ(J)*XME(J)
      XB = XD-XBAR(J)
      YB = YD-YBAR(J)
      ZB = ZD-ZBAR(J)
      D5 = (XI3(J)-XI1(J))**2+(ETA3(J)-ETA1(J))**2
      D6 = (XI4(J)-XI2(J))**2+(ETA4(J)-ETA2(J))**2
      D7 = AMAX1(D5,D6)
    3 XI = XLX*XB+XMX*YB+XNX*ZB
      ETA = XLE(J)*XB+XME(J)*YB+XNE(J)*ZB
      ZETA = XLZ(J)*XB+XMZ(J)*YB+XNZ(J)*ZB
      RO = XI**2+ETA**2+ZETA**2
      TJ = RO/D7
      IF(TJ.LT.6.0) GO TO 5
    4 SJ = .5*(XI3(J)-XI1(J))*(ETA2(J)-ETA4(J))/(RO*SQRT(RO))
      VXI = SJ*XI
      VETA = SJ*ETA
      VZETA = SJ*ZETA
      GO TO 90
    5 DO 6  I=1,4
      RJ(I) = SQRT((XI-XIK(J,I))**2+(ETA-ETAK(J,I))**2+ZETA**2)
      EJ(I) = ZETA**2+(XI-XIK(J,I))**2
      HJ(I) = (ETA-ETAK(J,I))*(XI-XIK(J,I))
      I1 = I+1
      IF(I.EQ.4) I1 = 1
      TRM1 = XIK(J,I1)-XIK(J,I)
      IF(TRM1.EQ.0.0) TRM1 = 1.0E-6
      EMJ(I) = (ETAK(J,I1)-ETAK(J,I))/TRM1
    6 CONTINUE
      VXI = 0.0
      VETA = 0.0
      VZETA = 0.0
    7 DO 9  I=1,4
      I1 = I+1
      IF(I.EQ.4) I1 = 1
      TRM1 = (RJ(I)+RJ(I1)-DDJ(J,I))/(RJ(I)+RJ(I1)+DDJ(J,I))
      TRM1 = ALOG(TRM1)
```

```
      TRM2 = (ETAK(J,I1)-ETAK(J,I))/DDJ(J,I)
      TRM3 = (XIK(J,I)-XIK(J,I1))/DDJ(J,I)
      VXI = VXI+TRM2*TRM1
      VETA = VETA+TRM3*TRM1
    8 IF(ZETA.EQ.0.0) GO TO 9
      TRM4 = ATAN((EMJ(I)*EJ(I)-HJ(I))/(ZETA*RJ(I)))
      TRM5 = ATAN((EMJ(I)*EJ(I1)-HJ(I1))/(ZETA*RJ(I1)))
      VZETA = VZETA+TRM4-TRM5
    9 CONTINUE
   90 VVX = XLX*VXI+XLE(J)*VETA+XLZ(J)*VZETA
      VVY = XMX*VXI+XME(J)*VETA+XMZ(J)*VZETA
      VVZ = XNX*VXI+XNE(J)*VETA+XNZ(J)*VZETA
      GO TO (10,11),NFLG
   10 VVVX = VVX
      VVVY = VVY
      VVVZ = VVZ
      YB = -YD-YBAR(J)
      NFLG = 2
      GO TO 3
   11 TRM = SIGX(J)*VXINF+SIGZ(J)*VZINF
      SUMU = SUMU+TRM*(VVVX+VVX)
      SUMV = SUMV+TRM*(VVVY-VVY)
      SUMW = SUMW+TRM*(VVVZ+VVZ)
   12 CONTINUE
   13 UFD = SUMU
      VFD = SUMV
      WFD = SUMW
      RETURN
      END
```

```
$IBFTC ZZVLCT  LIST,REF
C     WAKE VORTICITY CALCULATION PROGRAM - SUBROUTINE VLCTY
      SUBROUTINE VLCTY
      COMMON X(340,4),Y(340,4),Z(340,4),U(340,4),V(340,4),W(340,4),
     1       GAMA(340,4),SEG(340,4),GAMB1(100),A1(100),A(340,4),PI,RAD,
     2        VMP,XMCL,XMSL,NB,NRW,NA,NW,NOPT,NTAPE,NPRINT,NDVCH,PSIO,
     3        XMU,XLAM,ALPHAT,PINT,PSIF,XNA,DPSI,NB1,NW1,XNW,XNB,TPNB,
     4        SAT,CAT,C1,C2,PSI,TPI,XI,T1,T2,XJ,NPS,JPS,IPS,IPS1,XXX,YYY,
     5        ZZZ,IST,IND,IFLG,SIG1,SIG2,SIG3,GGG,DEN,XNU1,XNU2,XNU3,IR1,
     6        XMX,XMY,XMZ,SIG4,SIG5,RR,SQI1,SF,SQI,SG,BF,LPS,SEG1,SEG2,
     7        SUM,LNCT,XIPT(400),YIPT(400),ZIPT(400),VX(400),VY(400),
     8        VZ(400),VXF(400),VYF(400),VZF(400),NXPT,NAB,FACTR,RB
    1 DO 7  I=1,NXPT
      VX(I) = 0.0
      VY(I) = 0.0
      VZ(I) = 0.0
    2 DO 5  J=1,NB1
      SIG2 = SQRT((XIPT(I)-X(1,J))**2+(YIPT(I)-Y(1,J))**2+(ZIPT(I)-
     1        Z(1,J))**2)
    3 DO 4  K=1,NW
      SIG1 = SIG2
      SIG2 = SQRT((XIPT(I)-X(K+1,J))**2+(YIPT(I)-Y(K+1,J))**2+(ZIPT(I)-
     1        Z(K+1,J))**2)
      SEGSQ = SEG(K,J)**2
      HM1 = SIG1**2+SIG2**2
      IF(HM1.GT.SEGSQ)GO TO 30
      HM2 = .25*((SIG1+SIG2)**2-SEGSQ)*(SEGSQ-(SIG1-SIG2)**2)/SEGSQ
      IF(HM2.GT.A(K,J)**2)GO TO 30
      GGG = GAMA(K,J)/SEG(K,J)
      GO TO 31
   30 GGG = GAMA(K,J)*(SIG1+SIG2)/(SIG1*SIG2*((SIG1+SIG2)**2-SEGSQ))
   31 XNU1 = (YIPT(I)-Y(K,J))*(Z(K,J)-Z(K+1,J))-(ZIPT(I)-Z(K,J))*
     1        (Y(K,J)-Y(K+1,J))
      XNU2 = (ZIPT(I)-Z(K,J))*(X(K,J)-X(K+1,J))-(XIPT(I)-X(K,J))*
     1        (Z(K,J)-Z(K+1,J))
      XNU3 = (XIPT(I)-X(K,J))*(Y(K,J)-Y(K+1,J))-(YIPT(I)-Y(K,J))*
     1        (X(K,J)-X(K+1,J))
      VX(I) = VX(I)+XNU1*GGG
      VY(I) = VY(I)+XNU2*GGG
      VZ(I) = VZ(I)+XNU3*GGG
    4 CONTINUE
    5 CONTINUE
      SIG1 = SQRT(XIPT(I)**2+YIPT(I)**2+ZIPT(I)**2)
      DO 6  L=1,NB
      LPS = MOD(NPS+(NA*(L-1))/NB+NAB,NA)
      IF(LPS.EQ.0) LPS = NA
      XNU1 = -YIPT(I)*Z(1,L)+ZIPT(I)*Y(1,L)
      XNU2 = -ZIPT(I)*X(1,L)+XIPT(I)*Z(1,L)
      XNU3 = -XIPT(I)*Y(1,L)+YIPT(I)*X(1,L)
      SIG2 = SQRT((XIPT(I)-X(1,L))**2+(YIPT(I)-Y(1,L))**2+(ZIPT(I)-
     1        Z(1,L))**2)
      DEN = SIG1*SIG2*((SIG1+SIG2)**2-1.0)
      IF(DEN.EQ.0.0) DEN = .001
      GGG = GAMB1(LPS)*(SIG1+SIG2)/DEN
```

```
      VX(I) = VX(I)+XNU1*GGG
      VY(I) = VY(I)+XNU2*GGG
      VZ(I) = VZ(I)+XNU3*GGG
    6 CONTINUE
      VX(I) = VX(I)+XMCL+VXF(I)
      VY(I) = VY(I)+VYF(I)
      VZ(I) = VZ(I)-XMSL+VZF(I)
    7 CONTINUE
      RETURN
      END
```

APPENDIX II

OPERATIONAL INFORMATION FOR THE MODIFIED SUPPLEMENTAL FUSELAGE PROGRAM

This program is written completely in FORTRAN IV.

INPUTS

CARD 1	NPTS:	Number of fuselage elements N_p
	NPRNT:	Number of B_{ij} coefficients to be printed; i. e. NPRNT = (NPTS) (NPTS).
	NIT:	Maximum number of iterations to be allowed in solving the equations (in case of divergence of the iterations).
	NOPT:	$\neq 0$; indicates nonuniform flow option. $= 0$; indicates uniform flow option.
	NDVCH:	Not used.
CARD 2, 3, ...,	(2)(NPTS) + 1	x_{11}, y_{11}, z_{11} ; x_{21}, y_{21}, z_{21}, x_{31}, y_{31}, z_{31} ; x_{41}, y_{41}, z_{41}, x_{12}, y_{12}, z_{12} ; x_{22}, y_{22}, z_{22}, etc.

A listing of the program is given on the pages which follow.

```
$IBFTC ZZHFPF  LIST,REF
C     CALCULATION OF POTENTIAL FLOW ABOUT A HELICOPTER FUSELAGE
      COMMON X1(100),X2(100),X3(100),X4(100),Y1(100),Y2(100),Y3(100),
     1       Y4(100),Z1(100),Z2(100),Z3(100),Z4(100),XBAR(100),YBAR(100)
     2       ,ZBAR(100),AMTX(3,4),XPT(4),YPT(4),ZPT(4),XI1(100),XI2(100)
     3       ,XI3(100),XI4(100),ETA1(100),ETA2(100),ETA3(100),ETA4(100),
     4       ZETA1(100),ZETA2(100),ZETA3(100),ZETA4(100),XLX(100),
     5       XMX(100),XNX(100),XLF(100),XMF(100),XNF(100),XLZ(100),
     6       XMZ(100),XNZ(100),RIJ(4),EIJ(4),HIJ(4),D1(100),D2(100),
     7       D3(100),D4(100),B(100,102),SIGX(100),SIGZ(100),NPTS,NDVCH,
     8       EPS,AN,BN,GN,AX,BX,GX,AE,BE,GE,CX,CE,CZ,D5,D6,D7,SJ,NFLG,
     9       EM1,EM2,EM3,EM4,XPP,YPP,ZPP,YRPP,XIIJ,ETAIJ,ZETAIJ,RO,RI
      COMMON XIRIJ,ETARIJ,ZETRIJ,TIJ,TRIJ,VXI,VETA,VZETA,TMP1,TMP2,TMP3,
     1       TMP4,TMP,VX,VY,VZ,AIJ,ARIJ,N1,N2,NPRNT,MPRNT,NIT
      DIMENSION X(100,4),Y(100,4),Z(100,4),XIK(100,4),ETAK(100,4),
     1          ZETAK(100,4)
      EQUIVALENCE (X,X1),(Y,Y1),(Z,Z1),(XIK,XI1),(ETAK,ETA1),(ZETAK,
     1            ZETA1)
      DIMENSION VXM(100),VZM(100)
    1 CALL CLEAR(X1,NIT)
      READ 1000,NPTS,NPRINT,NIT,NOPT,NDVCH,EPS
 1000 FORMAT(5I6,F6.1)
      READ 1001,((X(I,J),Y(I,J),Z(I,J),J=1,4),I=1,NPTS)
 1001 FORMAT(6F12.5)
      IF (NOPT.NE.0) READ 1001,(VXM(I),VZM(I),I=1,NPTS)
      CALL DVDCHK(NDVCH)
    2 DO 18  I=1,NPTS
      AN = (Y4(I)-Y2(I))*(Z3(I)-Z1(I))-(Z4(I)-Z2(I))*(Y3(I)-Y1(I))
      BN = (Z4(I)-Z2(I))*(X3(I)-X1(I))-(X4(I)-X2(I))*(Z3(I)-Z1(I))
      GN = (X4(I)-X2(I))*(Y3(I)-Y1(I))-(Y4(I)-Y2(I))*(X3(I)-X1(I))
      XBAR(I) = (X1(I)+X2(I)+X3(I)+X4(I))/4.0
      YBAR(I) = (Y1(I)+Y2(I)+Y3(I)+Y4(I))/4.0
      ZBAR(I) = (Z1(I)+Z2(I)+Z3(I)+Z4(I))/4.0
    3 IF(AN.NE.0.0) GO TO 8
      IF(BN.NE.0.0) GO TO 8
      IF(GN.NE.0.0) GO TO 6
    4 WRITE(6,1002)I,(X(I,J),Y(I,J),Z(I,J),J=1,4)
 1002 FORMAT(40H BAD SET OF POINTS FOR QUADRILATERAL NO. I5/4(3F9.4,5X))
      GO TO 1
    6 DO 7  J=1,4
      XPT(J) = X(I,J)-XBAR(I)
      YPT(J) = Y(I,J)-YBAR(I)
      ZPT(J) = 0.0
    7 CONTINUE
      GO TO 14
    8 AMTX(1,1) = AN
      AMTX(1,2) = BN
      AMTX(1,3) = GN
      AMTX(2,1) = BN
      AMTX(2,2) = -AN
      AMTX(2,3) = 0.0
      IF(AN.NE.0.0) GO TO 11
    9 AMTX(3,1) = 0.0
      AMTX(3,2) = GN
```

```
      AMTX(3,3) = -BN
      DO 10  J=1,4
      AMTX(1,4) = BN*YBAR(I)+GN*ZBAR(I)
      AMTX(2,4) = BN*X(I,J)
      AMTX(3,4) = GN*Y(I,J)-BN*Z(I,J)
      CALL SIMSOL (AMTX,3,3)
      XPT(J) = AMTX(1,4)-XBAR(I)
      YPT(J) = AMTX(2,4)-YBAR(I)
      ZPT(J) = AMTX(3,4)-ZBAR(I)
   10 CONTINUE
      GO TO 14
   11 AMTX(3,1) = GN
      AMTX(3,2) = 0.0
      AMTX(3,3) = -AN
   12 DO 13  J=1,4
      AMTX(1,4) = AN*XBAR(I)+BN*YBAR(I)+GN*ZBAR(I)
      AMTX(2,4) = BN*X(I,J)-AN*Y(I,J)
      AMTX(3,4) = GN*X(I,J)-AN*Z(I,J)
      CALL SIMSOL(AMTX,3,3)
      XPT(J) = AMTX(1,4)-XBAR(I)
      YPT(J) = AMTX(2,4)-YBAR(I)
      ZPT(J) = AMTX(3,4)-ZBAR(I)
   13 CONTINUE
   14 AX = XPT(3)-XPT(1)
      BX = YPT(3)-YPT(1)
      GX = ZPT(3)-ZPT(1)
      AE = BN*GX-BX*GN
      BE = GN*AX-GX*AN
      GE = AN*BX-AX*BN
      CX = 1.0/SQRT(AX**2+BX**2+GX**2)
      CE = 1.0/SQRT(AE**2+BE**2+GE**2)
      CZ = 1.0/SQRT(AN**2+BN**2+GN**2)
      XLX(I) = CX*AX
      XMX(I) = CX*BX
      XNX(I) = CX*GX
      XLE(I) = CE*AE
      XME(I) = CE*BE
      XNE(I) = CE*GE
      XLZ(I) = CZ*AN
      XMZ(I) = CZ*BN
      XNZ(I) = CZ*GN
   16 DO 17  J=1,4
      XIK(I,J) = XLX(I)*XPT(J)+XMX(I)*YPT(J)+XNX(I)*ZPT(J)
      ETAK(I,J) = XLE(I)*XPT(J)+XME(I)*YPT(J)+XNE(I)*ZPT(J)
      ZETAK(I,J) = XLZ(I)*XPT(J)+XMZ(I)*YPT(J)+XNZ(I)*ZPT(J)
   17 CONTINUE
   18 CONTINUE
   19 DO 31  J=1,NPTS
      D1(J) = SQRT((XI2(J)-XI1(J))**2+(ETA2(J)-ETA1(J))**2)
      D2(J) = SQRT((XI3(J)-XI2(J))**2+(ETA3(J)-ETA2(J))**2)
      D3(J) = SQRT((XI4(J)-XI3(J))**2+(ETA4(J)-ETA3(J))**2)
      D4(J) = SQRT((XI1(J)-XI4(J))**2+(ETA1(J)-ETA4(J))**2)
      D5 = (XI3(J)-XI1(J))**2+(ETA3(J)-ETA1(J))**2
      D6 = (XI4(J)-XI2(J))**2+(ETA4(J)-ETA2(J))**2
```

```
      D7 = AMAX1(D5,D6)
      SJ = .5*(XI3(J)-XI1(J))*(ETA2(J)-ETA4(J))
      TRM = XI2(J)-XI1(J)
      IF(TRM.EQ.0.0)TRM = 1.0E-6
      EM1 = (ETA2(J)-ETA1(J))/TRM
      TRM = XI3(J)-XI2(J)
      IF(TRM.EQ.0.0)TRM = 1.0E-6
      EM2 = (ETA3(J)-ETA2(J))/TRM
      TRM = XI4(J)-XI3(J)
      IF(TRM.EQ.0.0)TRM = 1.0E-6
      EM3 = (ETA4(J)-ETA3(J))/TRM
      TRM = XI1(J)-XI4(J)
      IF(TRM.EQ.0.0)TRM = 1.0E-6
      EM4 = (ETA1(J)-ETA4(J))/TRM
   20 DO 30  I=1,NPTS
      NFLG = 1
      XPP = XBAR(I)-XBAR(J)
      YPP = YBAR(I)-YBAR(J)
      ZPP = ZBAR(I)-ZBAR(J)
      YRPP = -YBAR(I)-YBAR(J)
      XIIJ = XLX(J)*XPP+XMX(J)*YPP+XNX(J)*ZPP
      ETAIJ = XLE(J)*XPP+XME(J)*YPP+XNE(J)*ZPP
      ZETAIJ = XLZ(J)*XPP+XMZ(J)*YPP+XNZ(J)*ZPP
      XIRIJ = XLX(J)*XPP+XMX(J)*YRPP+XNX(J)*ZPP
      ETARIJ = XLE(J)*XPP+XME(J)*YRPP+XNE(J)*ZPP
      ZETRIJ = XLZ(J)*XPP+XMZ(J)*YRPP+XNZ(J)*ZPP
      R1 = XIRIJ**2+ETARIJ**2+ZETRIJ**2
      R0 = XIIJ**2+ETAIJ**2+ZETAIJ**2
      TIJ = R0/D7
      TRIJ = R1/D7
   21 IF(I.NE.J) GO TO 22
      VXI = 0.0
      VETA = 0.0
      VZETA = 6.2831853072
      GO TO 27
   22 IF(TIJ.GT.6.0) GO TO 26
   23 DO 24  K=1,4
      RIJ(K) = SQRT((XIIJ-XIK(J,K))**2+(ETAIJ-ETAK(J,K))**2+ZETAIJ**2)
      EIJ(K) = ZETAIJ**2+(XIIJ-XIK(J,K))**2
      HIJ(K) = (ETAIJ-ETAK(J,K))*(XIIJ-XIK(J,K))
   24 CONTINUE
      TMP1 = ALOG((RIJ(1)+RIJ(2)-D1(J))/(RIJ(1)+RIJ(2)+D1(J)))/D1(J)
      TMP2 = ALOG((RIJ(2)+RIJ(3)-D2(J))/(RIJ(2)+RIJ(3)+D2(J)))/D2(J)
      TMP3 = ALOG((RIJ(3)+RIJ(4)-D3(J))/(RIJ(3)+RIJ(4)+D3(J)))/D3(J)
      TMP4 = ALOG((RIJ(4)+RIJ(1)-D4(J))/(RIJ(4)+RIJ(1)+D4(J)))/D4(J)
      VXI = (ETA1(J)-ETA2(J))*TMP1+(ETA2(J)-ETA3(J))*TMP2+(ETA3(J)-
     1      ETA4(J))*TMP3+(ETA4(J)-ETA1(J))*TMP4
      VXI = -VXI
      VETA = (XI1(J)-XI2(J))*TMP1+(XI2(J)-XI3(J))*TMP2+(XI3(J)-XI4(J))*
     1       TMP3+(XI4(J)-XI1(J))*TMP4
      IF(ZETAIJ.NE.0.0) GO TO 25
      VZETA = 0.0
      GO TO 27
   25 VZETA = ATAN((EM1*EIJ(1)-HIJ(1))/(ZETAIJ*RIJ(1)))-ATAN((EM1*EIJ(2)
```

```
     1          -HIJ(2))/(ZETAIJ*RIJ(2)))+ATAN((EM2*EIJ(2)-HIJ(2))/(ZETAIJ
     2          *RIJ(2)))-ATAN((EM2*EIJ(3)-HIJ(3))/(ZETAIJ*RIJ(3)))+
     3          ATAN((EM3*EIJ(3)-HIJ(3))/(ZETAIJ*RIJ(3)))-ATAN((EM3*EIJ(4)
     4          -HIJ(4))/(ZETAIJ*RIJ(4)))+ATAN((EM4*EIJ(4)-HIJ(4))/(ZETAIJ
     5          *RIJ(4)))-ATAN((EM4*EIJ(1)-HIJ(1))/(ZETAIJ*RIJ(1)))
      GO TO 27
   26 TMP = SJ/(RO*SQRT(RO))
      VXI = XIIJ*TMP
      VETA = ETAIJ*TMP
      VZETA = ZETAIJ*TMP
   27 VX = XLX(J)*VXI+XLE(J)*VETA+XLZ(J)*VZETA
      VY = XMX(J)*VXI+XME(J)*VETA+XMZ(J)*VZETA
      VZ = XNX(J)*VXI+XNE(J)*VETA+XNZ(J)*VZETA
      GO TO (28,29),NFLG
   28 AIJ = XLZ(I)*VX+XMZ(I)*VY+XNZ(I)*VZ
      NFLG = 2
      XIIJ = XIRIJ
      ETAIJ = ETARIJ
      ZETAIJ = ZETRIJ
      RO = R1
      TIJ = TRIJ
      GO TO 22
   29 ARIJ = XLZ(I)*VX-XMZ(I)*VY+XNZ(I)*VZ
      B(I,J) = AIJ+ARIJ
   30 CONTINUE
   31 CONTINUE
      N1 = NPTS+1
      N2 = NPTS+2
   32 DO 33  I=1,NPTS
      IF (NOPT.NE.0) GO TO 320
      B(I,N1) = -XLZ(I)
      B(I,N2) = -XNZ(I)
      GO TO 33
  320 B(I,N1) = -XLZ(I)*VXM(I)-XNZ(I)*VZM(I)
      B(I,N2) = B(I,N1)
   33 CONTINUE
      IF(NPRNT.EQ.0) GO TO 38
      MPRINT = MIN0(NPRNT,NPTS)
   34 DO 37  I=1,MPRINT,8
      I8 = MIN0(I+7,MPRINT)
      WRITE(6,1003)I,I8,MPRINT
 1003 FORMAT(1H1,44X42HPOTENTIAL FLOW ABOUT A HELICOPTER FUSELAGE //50X
     1          11HBIJ FOR J =I3,2H -I3,9H , I = 1-I3/)
   35 DO 36  J=1,MPRINT
      WRITE(6,1004)(B(J,K),K=I,I8)
 1004 FORMAT(8E16.5)
   36 CONTINUE
   37 CONTINUE
   38 CALL SIMEQ
      CALL OUTPUT
      GO TO 1
      END
```

```
$IBFTC ZZSMEQ  LIST,REF
C     CALCULATION OF POTENTIAL FLOW ABOUT A HELICOPTER FUSELAGE - SIMEQ
      SUBROUTINE SIMEQ
      COMMON X1(100),X2(100),X3(100),X4(100),Y1(100),Y2(100),Y3(100),
     1       Y4(100),Z1(100),Z2(100),Z3(100),Z4(100),XBAR(100),YBAR(100)
     2       ,ZBAR(100),AMTX(3,4),XPT(4),YPT(4),ZPT(4),XI1(100),XI2(100)
     3       ,XI3(100),XI4(100),ETA1(100),ETA2(100),ETA3(100),ETA4(100),
     4       ZETA1(100),ZETA2(100),ZETA3(100),ZETA4(100),XLX(100),
     5       XMX(100),XNX(100),XLE(100),XME(100),XNE(100),XLZ(100),
     6       XMZ(100),XNZ(100),RIJ(4),EIJ(4),HIJ(4),D1(100),D2(100),
     7       D3(100),D4(100),B(100,102),SIGX(100),SIGZ(100),NPTS,NDVCH,
     8       EPS,AN,BN,GN,AX,BX,GX,AE,BE,GE,CX,CE,CZ,D5,D6,D7,SJ,NFLG,
     9       EM1,EM2,EM3,EM4,XPP,YPP,ZPP,YRPP,XIIJ,ETAIJ,ZETAIJ,RO,R1
      COMMON XIRIJ,ETARIJ,ZETRIJ,TIJ,TRIJ,VXI,VETA,VZETA,TMP1,TMP2,TMP3,
     1       TMP4,TMP,VX,VY,VZ,AIJ,ARIJ,N1,N2,NPRNT,MPRNT,NIT
      DIMENSION X(100,4),Y(100,4),Z(100,4),XIK(100,4),ETAK(100,4),
     1          ZETAK(100,4)
      EQUIVALENCE (X,X1),(Y,Y1),(Z,Z1),(XIK,XI1),(ETAK,ETA1),(ZETAK,
     1          ZETA1)
      DIMENSION U1(100),U2(100)
    1 JS = N1
      EPS1 = 100.0*EPS
    2 IT = 0
    3 DO 6  I=1,NPTS
      U1(I) = 0.0
    4 DO 5  J=1,NPTS
      TERM = -B(I,J)/B(I,I)
      IF(I.EQ.J) TERM = 0.0
      U1(I) = U1(I)+TERM*U1(J)
    5 CONTINUE
      U1(I) = U1(I)+B(I,JS)/B(I,I)
    6 CONTINUE
    7 DO 10  I=1,NPTS
      IF(U2(I).NE.0.0)GO TO 8
      TMP = ABS(U2(I)-U1(I))
      GO TO 9
    8 TMP = ABS((U2(I)-U1(I))/U2(I))
    9 IF(TMP.GT.EPS) GO TO 15
   10 CONTINUE
      IF(JS.EQ.N2) GO TO 13
   11 DO 12  I=1,NPTS
      SIGX(I) = U1(I)
   12 CONTINUE
      JS = N2
      GO TO 2
   13 DO 14  I=1,NPTS
      SIGZ(I) = U1(I)
   14 CONTINUE
      RETURN
   15 IT = IT+1
      IF (IT.GE.NIT) GO TO 18
   16 DO 17  I=1,NPTS
      U2(I) = U1(I)
   17 CONTINUE
```

```
      GO TO 3
   18 IF(JS.EQ.N2) GO TO 19
      WRITE(6,1000) EPS1,NIT
 1000 FORMAT (5X48HEQUATIONS FOR SIGMA X DID NOT CONVERGE TO WITHIN F7.4
     1       ,12H PER CENT IN I6,11H ITERATIONS )
      GO TO 11
   19 WRITE(6,1001) EPS1,NIT
 1001 FORMAT( 5X48HEQUATIONS FOR SIGMA Z DID NOT CONVERGE TO WITHIN F7.4
     1       ,12H PERCENT IN I6,11H ITERATIONS )
      GO TO 13
      END
```

```
$IBFTC ZZOUT   LIST,REF
C     CALCULATION OF POTENTIAL FLOW ABOUT A HELICOPTER FUSELAGE - OUTPUT
      SUBROUTINE OUTPUT
      COMMON XI(100),X2(100),X3(100),X4(100),Y1(100),Y2(100),Y3(100),
     1         Y4(100),Z1(100),Z2(100),Z3(100),Z4(100),XBAR(100),YBAR(100)
     2         ,ZBAR(100),AMTX(3,4),XPT(4),YPT(4),ZPT(4),XI1(100),XI2(100)
     3         ,XI3(100),XI4(100),ETA1(100),ETA2(100),ETA3(100),ETA4(100),
     4         ZETA1(100),ZETA2(100),ZETA3(100),ZETA4(100),XLX(100),
     5         XMX(100),XNX(100),XLE(100),XME(100),XNE(100),XLZ(100),
     6         XMZ(100),XNZ(100),RIJ(4),EIJ(4),HIJ(4),D1(100),D2(100),
     7         D3(100),D4(100),B(100,102),SIGX(100),SIGZ(100),NPTS,NDVCH,
     8         EPS,AN,BN,GN,AX,BX,GX,AE,BE,GE,CX,CE,CZ,D5,D6,D7,SJ,NFLG,
     9         EM1,EM2,EM3,EM4,XPP,YPP,ZPP,YRPP,XIIJ,ETAIJ,ZETAIJ,R0,RI
      COMMON XIRIJ,ETARIJ,ZETRIJ,TIJ,TRIJ,VXI,VETA,VZETA,TMP1,TMP2,TMP3,
     1         TMP4,TMP,VX,VY,VZ,AIJ,ARIJ,N1,N2,NPRNT,MPRNT,NIT
      DIMENSION X(100,4),Y(100,4),Z(100,4),XIK(100,4),ETAK(100,4),
     1         ZETAK(100,4)
      EQUIVALENCE (X,X1),(Y,Y1),(Z,Z1),(XIK,XI1),(ETAK,ETA1),(ZETAK,
     1         ZETA1)
      DIMENSION T1(100)
    1 PUNCH 1000,NPTS
 1000 FORMAT(32HZZ    HELICOPTER FUSELAGE PROGRAM I6,22H POINTS - HARVEY
     1SELIB,12X2HZZ )
      I1 = 1
      I2 = 2
      I3 = 3
      I4 = 4
    2 DO 3  I=1,NPTS
      PUNCH 1001, XBAR(I),YBAR(I),ZBAR(I),SIGX(I),SIGZ(I),ZERO,I,I1
      PUNCH 1001,XI1(I),XI2(I),XI3(I),XI4(I),ETA1(I),ETA2(I),I,I2
      PUNCH 1001,ETA3(I),ETA4(I),D1(I),D2(I),D3(I),D4(I),I,I3
      PUNCH 1001,XLE(I),XME(I),XNE(I),XLZ(I),XMZ(I),XNZ(I),I,I4
 1001 FORMAT(6E12.5,I6,I2)
    3 CONTINUE
      NPAGE = NPTS/50
      IF(NPAGE*50.LT.NPTS)NPAGE = NPAGE+1
    4 DO 5  I=1,NPAGE
      I1 = 50*(I-1)+1
      I2 = MIN0(I1+49,NPTS)
      WRITE(6,1002)(SIGX(J),SIGZ(J),XLZ(J),XMZ(J),XNZ(J),J=I1,I2)
 1002 FORMAT(1H1,44X42HPOTENTIAL FLOW ABOUT A HELICOPTER FUSELAGE //18X
     1         7HSIGMA X,15X7HSIGMA Z,12X11HLAMBDA ZETA, 14X7HMU ZETA,15X
     2         7HNU ZETA / (F27.5,4E22.5) )
    5 CONTINUE
      PUNCH 1003
 1003 FORMAT(80H*******************************************************
     1*************************)
    6 DO 7  I=1,NPTS
      T1(I)=-YBAR(I)
    7 CONTINUE
      PUNCH 1004,(XBAR(I),YBAR(I),ZBAR(I),XBAR(I),T1(I),ZBAR(I),
     1         I=1,NPTS)
 1004 FORMAT(9F8.3)
      RETURN
      END
```

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Cornell Aeronautical Laboratory, Inc.
Buffalo, New York 14221

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Theoretical Prediction of the Flow in the Wake of a Helicopter Rotor
Addendum - Effects Due to a Fuselage in a Constant, Nonuniform Flow

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Report Addendum January - May 1966

5. AUTHOR(S) *(Last name, first name, initial)*

Crimi, Peter
Trenka, Andrew R.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1966	43	2

8a. CONTRACT OR GRANT NO.

DA 30-069-AMC-645(R)

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

BB-1994-S-3

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES Parts 1 and 2 (BB-1994-S-1, 2) discusses the problem formulation, computational results and digital programs.

12. SPONSORING MILITARY ACTIVITY

U. S. Army
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland

13. ABSTRACT

This addendum reports the results of an extension to the study to determine the time varying flow in the vicinity of a helicopter rotor in hovering and forward flight and having a fuselage immersed in the rotor wake. The purpose of this extension was to compare two models representing the fuselage. The first fuselage model, reported in Parts 1 and 2, was based upon the assumption that the fuselage was immersed in a constant and uniform flow; the second model was based upon the assumption that the fuselage was immersed in a constant but non-uniform flow.

Comparisons between the two models are presented. Program listings and operational information related to the programs are given in the appendices.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Helicopter Rotor Rotor Flow Field Aerodynamics Fluid Dynamics Digital Computations						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ______."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ______."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ______."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.