[redacted]

NASA Technical Memorandum 104139

AVSCOM Technical Report 91-B-016

P-28

USER'S GUIDE FOR A "FLAT WAKE" ROTOR INFLOW/WAKE VELOCITY PREDICTION CODE, "DOWN"

[redacted]

(NASA-TM-104139) USER'S GUIDE FOR A FLAT WAKE ROTOR INFLOW/WAKE VELOCITY PREDICTION CODE, DOWN (NASA. Langley Research Center) 28 p

N94-32873

Unclas

G3/02 0011942

John C. Wilson

November 1991

Review for general release November 30, 1993

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

US ARMY AVIATION SYSTEMS COMMAND
AVIATION R&T ACTIVITY

SUMMARY

A computer code named DOWN has been created to implement a "flat wake theory" for the calculation of rotor inflow and wake velocities. The theory was developed by V.E. Baskin of the USSR. The code was developed at Princeton University under a Grant from the National Aeronautics and Space Administration (NASA). A brief description of the code methodology and instructions for its use are given. The code will be available from NASA's Computer Software Management and Information Center (COSMIC).

PROBLEM DEFINITION

The prediction of inflow to a helicopter rotor and the wake velocities below and behind are vital to the calculation of airloads on the rotor blades of a helicopter and the other components. Inflow and wake velocities affect the local angles-of-attack, and therefore the airloads, that the helicopter rotor and other components experience.

Many theories and analyses have been developed for prediction of both mean and time-varying inflow/wake patterns. Computer codes to implement these predictions have grown in complexity as the theories have grown in complexity. Rotor performance codes use inflow models that range from the assumption of uniform inflow to complex, time-varying, vortex filament, "free-wake" models. Even though computer processing time and associated costs have dramatically decreased in recent years, there is still need for codes which are simple, fast, and accurate for preliminary design and analysis.

PROBLEM SOLUTION

A relatively simple analysis for predicting helicopter inflow/wake velocities called "flat wake theory" was implemented in a computer code, named DOWN. The analysis essentially treats the rotor wake geometry as rigid without interaction between induced velocities and wake structure. The flat wake theory for rotors was developed by V.E. Baskin of the USSR and is described in reference 1. An analytic scheme for implementing the theory is described in reference 2. The theory and associated assumptions and limitations are described in references 1, 2 and 3.

The computer coding of the scheme was implemented by Professors Howard C. Curtiss and Robert M. McKillip of Princeton University under National Aeronautics and Space Administration Research Grant No. NAG-1-1038 (Studies of Rotor Inflow Using a Flat Wake Theory Including Correlation with Experiment). The code will predict the three orthogonal incremental components of flow velocity (as ratios of incremental velocities to tip speed) at any point in any plane parallel or perpendicular to the rotor disk. The input to the code is quite simple and is entered interactively through the computer keyboard.

The predictive capability of the coded version of the theory has been correlated with flow velocity data of several sources. In general, the coded version of flat wake theory provides vertical inflow patterns similar to experimental patterns for helicopter flight speeds greater than approximately 60 knots (or, as indicated in reference 1, for rotor advance ratio greater than 0.15).

SYMBOLS

Symbols used in the code are bracketed.

C_T	(CT)	rotor thrust coefficient, $T/[\rho_a (\Omega R)^2 \pi R^2]$
G		circulation "normalizing factor"
r		rotor blade station, ft
R		rotor radius, ft
T		rotor thrust, lbs
v_o		momentum theory value of average induced velocity, fps
$\overline{v}_o$		v_o normalized with tip speed i.e. v_o/V_T
V		wind or flight speed, ft/sec
V_t		rotor tip speed, ΩR, ft/sec
v_x		increment in longitudinal velocity due to rotor inflow/wake, positive forward, ft/sec
v_y		increment in vertical velocity due to rotor inflow/wake, positive upward, ft/sec
v_z		increment in sidewash velocity due to rotor inflow/wake, positive to the right when view is in direction of flight, ft/sec
x,y,z	(X,Y,Z)	Cartesian coordinate system centered in rotor hub as shown in figure 1, (convention of ref. 1 whereby x-z is the tip path plane and y upward, positive sense is as for the velocity increments, v_x, v_y, and v_z) non-dimensionalized with rotor radius, /R
Γ	(GAMMA)	blade circulation, ft^2/sec
	(GAMSIN)	factor for azimuthal variation of circulation as a function of sine of azimuth angle
	(GAMCOS)	factor for azimuthal variation of circulation as a function of cosine of azimuth angle
	(GAMFACT)	ratio of azimuthally affected circulation to assumed "average" circulation
$\Delta\lambda$	(DLAM)	increment in rotor inflow ratio, v_y/V_t
$\Delta\mu$	(DMU)	increment in rotor advance ratio, v_x/V_t
μ	(VBAR)	rotor advance ratio, V/V_t
$\Delta\eta$	(DNU)	increment in sidewash velocity component, v_z/V_t
ρ	(RBAR)	rotor blade station, r/R
ρ_a		air density, $slugs/ft^3$
ψ	(PSI)	blade azimuth position, deg
Ω		rotor rotational speed, rad/sec

CODE METHODOLOGY

The development of the theory began with the consideration of a rotor vortex system which is formed mainly by shed vorticity leaving the rotor blades. The generation of this vorticity is a function of the radial variation of blade lift (i.e. "circulation"). As the horizontal flight speed increases the vortex column below the rotor deflects more to the rear. At sufficiently high speeds, the column becomes practically flat resulting in a relatively undistorted sheet, or ribbon of vorticity transported downstream at the free stream velocity. Vortices that constitute that ribbon (essentially attributable to the incremental changes in the radial variation of circulation) spring from all of the rotor blade sections. Thus, cycloidal patterns of vorticity of various shapes and dimensions are formed. In the analytical scheme and code these cycloidal vortices are represented by an equivalent rectilinear vortex system. References 2 and 3 offer a full description of the treatment of vorticity.

In the computation of resultant induced velocities the contribution of lateral and longitudinal vortices are computed separately and then summed. An important assumption in the theory development is that the circulation varies only with radius and that the contributions of azimuthal variations of vorticity are relatively small. However, a means for accounting for azimuthal variation is provided in the code and is described in Appendix A.

The code is written in FORTRAN-77 and can be compiled on the MicroSoft Version 5 compiler. There are 500 lines of code which can be reduced to less than 350 lines if a user wishes to limit the calculation to a single field point velocity calculation (i.e. to be used as a subroutine, perhaps). The code is easily modified to the requirements of the user and requires no special hardware or software environment. A listing is given in Appendix B.

The operation of the code begins with a selection of circulation ($\approx$ lift) distribution on the rotor blade as a function of radius. The source code DOWN is compiled with one of three subroutines describing representative radial variations of circulation. The subroutines for several circulations as functions of blade radial station are identified as GA1, GA2, or GA3 and shown in Appendix C. GA1 is a linear distribution as shown on fig. 3.3 of reference 1. GA2 is a parabolic distribution described on page 56 of reference 1. GA3 is similar to equation 3.34 of reference 1. Within the code, the circulation is normalized so that the output velocities are normalized with tip speed. The "normalization" allows for any circulation pattern a user of the code may wish to try and is described in Appendix D.

The organization of the code begins with the declaration of a parameter, NSEG, which is equated to the value of ten. NSEG is the number of blade segments for which incremental circulation will be calculated. NSEG can be increased for greater accuracy though there will be an associated increase in time required for calculation. Next, the user identifies the output file name and circulation subroutine called for in the compilation. The "normalization" factor for circulation is then calculated. Circulation (i.e. GAMMA) for the number of blade segments chosen is calculated and normalized. With that, the increment (or decrement) in circulation between segments is determined. These will be used to calculate the velocity increments after the integration factors (K, L, M, N, which are described in reference 1) are completed.

Before these integration factors are determined, in the main part of the program, the user chooses the location and type of velocity calculation. After the key input of circulation factors, advance ratio, and rotor thrust coefficient are given to the code, the dimensions required by the choice of calculation are input. These are the location of a plane, the initial, end, and incremental dimensions. That concludes the input and the code begins the primary loop ("DO 50") for producing the integral factors K, L, M, and N. Before the leaving the loop the velocity increments effected by all blade segments are defined for a field point. These are then normalized i.e. converted to ratios of velocity increments divided by tip speed.

USER INSTRUCTIONS

The source code, DOWN (Appendix C), should be compiled with one of the chosen circulation subroutines (Appendix D) using a FORTRAN-77 compiler. The input to the executable file, then (from the keyboard in response to requests shown on the computer monitor), is :

a. A name for the output file.
b. Identification of circulation subroutine (GA1, GA2, GA3, or other).
c. Choice of plane of calculation (calculation option 1, 2, 3, 4, or 5).
   1. Longitudinal-vertical (x-y) plane at lateral position z.
   2. Lateral-vertical (z-y) plane at longitudinal position x.
   3. Horizontal (x-z) plane at vertical height y.
   4. Radial variation at various azimuths (horizontal plane at height y).
   5. Azimuthal variation at various radii (horizontal plane at height y).
d. Factors GAMSIN and GAMCOS accounting for azimuthal variation of circulation (described in Appendix A)
e. Advance ratio, μ.
f. Rotor thrust coefficient, C_T.

Depending, then, on the choice (c) of plane of calculation (see figure 1) desired, input will be requested as follows:

If 1 Lateral location of vertical plane, Z.
Initial height, YINIT.
Final height, YFINL.
Incremental height, DELTAY.
Initial longitudinal distance, XINIT.
Final longitidinal distance, XFINL.
Incremental longitudinal distance, DELTAX.

If 2 Longitudinal location of vertical plane, Y.
Initial lateral distance, ZINIT.
Final lateral distance, ZFINL.
Incremental lateral distance, DELTAZ.
Initial height, YINIT.
Final height, YFINL.
Incremental height, DELTAY.

If 3 Height of horizontal plane, Y.
Initial longitudinal distance, XINIT.
Final longitudinal distance, XFINL.
Incremental longitudinal distance, DELTAX.
Initial lateral distance, ZINIT.
Final lateral distance, ZFINL.
Incremental lateral distance, DELTAZ.

If 4 Height of horizontal plane, Y.
Initial radial distance, RINIT.
Final radial distance, RFINL.
Incremental radial distance, DELTAR.
Initial azimuth, AZINIT.
Final azimuth, AZFINL.
Incremental azimuth, DELTAP.

If 5 Height of horizontal plane, Y.
Initial azimuth, AZINIT.
Final azimuth, AZFINL.
Incremental azimuth, DELTAP.
Initial radial distance, RINIT.
Final radial distance, RFINL.
Incremental radial distance, DELTAR.

If "choice" c is 1, 2, or 3, then the coordinates X, Y, and Z of the field point are listed along with the incremental nondimensional velocities $\Delta\lambda$, $\Delta\mu$, and $\Delta\eta$ and the corresponding radial position, ρ, and azimuthal coordinate, ψ. The factor, GAMFACT, described in Appendix A, is listed as well.

If "choice" c is 4 or 5, then the radial coordinate, ρ, and azimuthal coordinate, ψ, are listed along with incremental nondimensional velocities $\Delta\lambda$, $\Delta\mu$, and $\Delta\eta$ and the factor, GAMFACT.

SAMPLE INPUT/OUTPUT

Input

Output file name,	DwnDemo
Circulation subroutine name used in compilation,	GA3
Calculation option,	5
GAMSIN,	1.5
GAMCOS,	1.12
Advance ratio, VBAR,	0.149
Rotor thrust coefficient, CT,	0.00630

Height, Y,	0.077
Initial azimuth, AZINIT,	0.
Final azimuth, AZFINL,	90.
Increment in azimuth, DELTAP	30.
Initial radius, RINIT,	0.2
Final radius, RFINL,	1.2
Increment in radius, DELTAR	0.2

Output

Output File, FNAME : DwnDemo
Circulation program: GA3
Radial variation at various Azimuths
at Height, Y: .077; VBAR: .149; CT: .00630; GAMSIN: 1.500; GAMCOS: 1.120

PSI	r/R	DLAM	DMU	DNU	GAMFACT
0.	.20	-.00025	.01301	.02746	1.0000
0.	.40	-.00197	.01757	.03128	1.0000
0.	.60	-.00784	.01987	.03249	1.0000
0.	.80	-.01658	.01896	.03256	1.0000
0.	1.00	-.02665	.00966	.03121	1.0000
0.	1.20	-.02191	.00168	.03003	1.0000
30.	.20	-.01812	.01172	.02903	.9007
30.	.40	-.03214	.01582	.02172	.9007
30.	.60	-.04297	.01789	.01444	.9007
30.	.80	-.05509	.01708	.00454	.9007
30.	1.00	-.06681	.00870	-.01249	.9007
30.	1.20	-.05932	.00151	-.03101	.9007
60.	.20	-.02333	.01098	.01562	.8437
60.	.40	-.03120	.01482	.00555	.8437
60.	.60	-.03757	.01676	-.00673	.8437
60.	.80	-.04084	.01600	-.02364	.8437
60.	1.00	-.03639	.00815	-.05727	.8437
60.	1.20	.04101	.00142	-.04796	.8437
90.	.20	-.02165	.01075	.00815	.8262
90.	.40	-.02520	.01452	-.00254	.8262
90.	.60	-.02417	.01642	-.01465	.8262
90.	.80	-.01640	.01565	-.02870	.8262
90.	1.00	.01875	.00776	-.03536	.8262
90.	1.20	.01472	.00138	-.00496	.8262
120.	.20	-.01896	.01098	.00426	.8437
120.	.40	-.01900	.01482	-.00490	.8437
120.	.60	-.01400	.01676	-.01290	.8437

120.	.80	-.00341	.01600	-.01890	.8437
120.	1.00	.01448	.00815	-.01276	.8437
120.	1.20	.00844	.00142	-.00197	.8437
150.	.20	-.01670	.01172	.00263	.9007
150.	.40	-.01506	.01582	-.00395	.9007
150.	.60	-.00914	.01789	-.00817	.9007
150.	.80	.00015	.01708	-.01009	.9007
150.	1.00	.01154	.00870	-.00582	.9007
150.	1.20	.00662	.00151	-.00093	.9007
180.	.20	-.01521	.01301	.00236	1.0000
180.	.40	-.01349	.01757	-.00147	1.0000
180.	.60	-.00763	.01987	-.00268	1.0000
180.	.80	.00111	.01896	-.00274	1.0000
180.	1.00	.01119	.00966	-.00139	1.0000
180.	1.20	.00644	.00168	-.00022	1.0000
210.	.20	-.01408	.01463	.00274	1.1242
210.	.40	-.01391	.01975	.00180	1.1242
210.	.60	-.00859	.02233	.00392	1.1242
210.	.80	.00087	.02132	.00585	1.1242
210.	1.00	.01289	.01086	.00375	1.1242
210.	1.20	.00748	.00189	.00062	1.1242
240.	.20	-.01231	.01602	.00320	1.2309
240.	.40	-.01597	.02162	.00468	1.2309
240.	.60	-.01296	.02445	.01134	1.2309
240.	.80	-.00244	.02334	.01805	1.2309
240.	1.00	.01725	.01189	.01287	1.2309
240.	1.20	.01028	.00207	.00203	1.2309
270.	.20	-.00847	.01657	.00365	1.2732
270.	.40	-.01794	.02238	.00469	1.2732
270.	.60	-.02184	.02530	.01519	1.2732
270.	.80	-.01656	.02411	.03120	1.2732
270.	1.00	.02242	.01195	.04074	1.2732
270.	1.20	.01809	.00212	.00578	1.2732
300.	.20	-.00150	.01602	.00605	1.2309
300.	.40	-.01444	.02162	.00155	1.2309
300.	.60	-.02797	.02445	.00850	1.2309
300.	.80	-.03789	.02334	.02437	1.2309
300.	1.00	-.03830	.01189	.06067	1.2309
300.	1.20	.04573	.00207	.05271	1.2309
330.	.20	.00468	.01463	.01560	1.1242
330.	.40	-.00105	.01975	.00464	1.1242
330.	.60	-.01545	.02233	.00044	1.1242
330.	.80	-.03255	.02132	.00243	1.1242

330.	1.00	-.04961	.01086	.01241	1.1242
330.	1.20	-.04605	.00189	.02620	1.1242

DESCRIPTION of OPERATING ENVIRONMENT

The source code "DOWN" was compiled with the Microsoft FORTRAN Version 5.0 on a Hewlett Packard VECTRA QS/20 (IBM compatible) computer. The Microsoft FORTRAN compiler operates on any IBM or IBM-compatible computer running MS-DOS Version 3.0 or later version. The operating system for the HP VECTRA was MS-DOS Version 4.0. The compiled program, DOWN, can then be run under DOS Version 2.1 or later. The VECTRA had a math coprocessor chip and with its clock rate of 20 mhz a velocity calculation at a point was approximately 4 seconds.

REFERENCES

1. Baskin, V.E.; Vil'dgrube, L.S.; Vozhdayev, Y.S.; and Maykapar, G.I.: *Theory of the Lifting Airscrew.* NASA TTF-823, Feb. 1976.

2. Stepniewski, W.Z.; and Keys, C.N.: *Rotary-Wing Aerodynamics, Vol. I--Basic Theories of Rotor Aerodynamics (with Application to Helicopters).* NASA CR-3082, Jan. 1979.

3. Vil'dgrube, L.S.: *Helicopters.* (FSTC-HT-659-85, translation), Moscow "Mashinostroyeniye", 1977.

APPENDIX A

The azimuth variables, GAMSIN and GAMCOS, offer a means to account for variability of circulation with azimuth. According to reference 1

$$\Gamma(\rho,\mu,\psi) = \Gamma(\rho) + \Delta\Gamma(\mu,\psi)$$

Though the theory is developed with $\Delta\Gamma(\mu,\psi) \approx 0.0$, azimuth dependency may be introduced as on page 95 of reference 1:

$$\Delta\Gamma(\mu,\psi) \approx -1.5\ \mu \sin\psi + 1.12\ \mu^2 (1 - \cos 2\psi)$$

In the code then:

$$\Delta\Gamma(\mu,\psi) \approx -(\mathrm{GAMSIN})(\mathrm{VBAR})\sin(\mathrm{PSI}) + (\mathrm{GAMCOS})(\mathrm{VBAR})(\mathrm{VBAR})(1 - \cos(2*\mathrm{PSI}))$$

where GAMSIN = 0.0 or 1.5

and GAMCOS = 0.0 or 1.12

In the output of the code an accounting for azimuth dependency is given as GAMFACT where:

$$\mathrm{GAMFACT} = (\ \Gamma(\rho) + \Delta\Gamma(\mu,\psi))\ /\ \Gamma(\rho)$$

APPENDIX B

PROGRAM DOWN

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       CALCULATION OF ROTOR DOWNWASH FIELD USING FLAT-WAKE THEORY.
C       PROGRAM ORIGINALLY WRITTEN BY M. HAGLUND, '91, PRINCETON UNIV.
C       MODIFIED BY R. MCKILLIP 7/90 FOR IBM-PC USAGE.
C       (ADDITIONAL MODIFICATIONS BY J. WILSON  10/90 -- 2/91)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       NSEG IS NUMBER OF ROTOR SEGMENTS
C
        IMPLICIT INTEGER (I-N)
        IMPLICIT REAL (A-H,O-Z)
        REAL J,K,KINF,L,LAMBDA,LINF,M,MINF,MU,N,NINF,NU
        PARAMETER (NSEG=10)
C
        REAL GAMMA(NSEG),DGAMMA(NSEG),DVYI(NSEG),DVZI(NSEG),DVXI(NSEG)
        CHARACTER*10 FNAME
        CHARACTER*10 CIRC
C
        DATA PI/3.141592654/
C
100     FORMAT(1X,79A)
        WRITE(*,100) ' ******************************************'
        WRITE(*,100) ' *    FLAT WAKE DOWNWASH CALCULATION      *'
        WRITE(*,100) ' ******************************************'
        D2R = PI/180.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       OPEN FILES FOR OUTPUT
C       UNIT 1 IS DATA; PARAMETERS ARE PRINTED ON TITLE LINE.
C
        WRITE(*,100) ' Enter filename for output data:'
        READ(*,'(A)') FNAME
        OPEN(1,FILE=FNAME)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        WRITE(1,8) FNAME
8       FORMAT(1X,' Output File,  FNAME :',10A)
        WRITE(*,100) ' Enter Circulation Program used (i.e. GA1, etc.) '
        READ(*,'(A)') CIRC
        WRITE(1,9) CIRC
9       FORMAT(1X,' Circulation program ',10A)
```

```
C
C       CIRCULATION GAMMA IS COMPUTED USING A FUNCTION OF THE FORM:
C       GAMMA = GA(RHO), WHICH IS COMPILED SEPARATELY AND LINKED TO
C       THIS CODE, AS APPROPRIATE FOR THE TARGET COMPUTER.
C
C       THIS SECTION COMPUTES BOTH THE CIRCULATION GAMMA(I) AND THE
C       INCREMENTAL CIRCULATION DGAMMA(I).
C
C       FIRST, CALCULATE THE NORMALIZATION FACTOR
C
        G = 0.0
        NSTEPS = 200
        BG = 0.0
        ED = 1.0
        DRHO = (ED-BG)/FLOAT(NSTEPS)
        RHO = BG
        RHO2 = RHO + DRHO
        GAMA = GA(RHO)
4       CONTINUE
        IF (RHO2.LT.ED) THEN
            GAMMAD = GA(RHO2)
            G = G + (GAMA*RHO) + (GAMMAD*RHO2)
            GAMA = GAMMAD
C
            RHO = RHO2
            RHO2 = RHO2 + DRHO
        GOTO 4
        ENDIF
        G = G * DRHO
C
C       NOW CALCULATE THE CIRCULATION AND NORMALIZE BY G
C
        DO 5 I=1,NSEG
            RHO = .5*(2*I-1)/FLOAT(NSEG)
            GAMMA(I) = GA(RHO)/G
5       CONTINUE
        GAMMA(1) = 0.0
C        THIS LINE HAS BEEN REPLACED BY THE DO LOOP THAT FOLLOWS.
C        GAMMA(11) = 0.0
C
C       ...AND CONVERT NORMALIZED CIRCULATION TO A STAIR-STEP
C      FUNCTION FOR THE CALCULATION.
C
```

```
      DO 6  I = 1, NSEG-1
         DGAMMA(I) = GAMMA(I) - GAMMA(I+1)
6     CONTINUE
      DGAMMA(NSEG) = GAMMA(NSEG)
CCCCCCCCC1-CCCCCCCC2-CCCCCCCC3-CCCCCCCC4-CCCCCCCC5-CCCCCCCC6- CC
C
C     ASK FOR LOCATION AND TYPE OF VELOCITY CALCULATION
C
7     CONTINUE
      WRITE(*,100) ' Enter Calculation Option:'
      WRITE(*,100) ' 1)  Longitudinal/Vertical plane at Lateral dist.'
      WRITE(*,100) ' 2)  Lateral/Vertical plane at Longitudinal dist.'
      WRITE(*,100) ' 3)  Horizontal plane at Vertical Height'
      WRITE(*,100) ' 4)  Radial variation at various Azimuths'
      WRITE(*,100) ' 5)  Azimuthal variation at various Radii'
      WRITE(*,100) ' ?: '
C
      READ(*,*) IDIR
      IF( (IDIR.LT.1).OR.(IDIR.GT.5) ) GO TO 7
     WRITE(*,100) ' Enter factors for circulation varying with az.'
      READ(*,*) GAMSIN,GAMCOS
C
      WRITE(*,100) ' Enter ADVANCE RATIO: '
      READ(*,*) VBAR
C
      WRITE(*,100) ' Enter ROTOR THRUST COEFFICIENT: '
      READ(*,*) CT
C
C-----INCREMENT ALONG X-DIRECTION (and Y) for long./vert. plane
C
      IF( IDIR.EQ.1 ) THEN
      WRITE(*,100) ' Enter lateral location of Vertical plane:'
      READ(*,*) Z
      WRITE(*,100) ' Enter initial, final, and delta Height'
      READ(*,*) YINIT,YFINL,DELTAY
      WRITE(*,100) ' Enter initial, final, and delta Long. dist.'
      READ(*,*) XINIT,XFINL,DELTAX
      WRITE(1,100) ' Longitudinal/Vertical plane at Lateral dist.'
      WRITE(1,10) Z,VBAR,CT,GAMSIN,GAMCOS
10    FORMAT(1X,'   at, Z:',F6.3,'; VBAR:',F5.3,';  CT:',
     >  F7.5,';                  GAMSIN:',F5.3,';  GAMCOS:',F5.3)
      WRITE(1,100) '   X     Y     Z     DLAM     DMU     DNU
     > GAMMA    R                PHI'
```

```
      WRITE(1,100) '--------- --------- --------- --------- --------- -
     >-------- --------- ---------'
      X = XINIT - DELTAX
      Y = YINIT
C
C-----INCREMENT ALONG Y-DIRECTION (and Z) for lateral/vertical plane
C
       ELSE IF ( IDIR.EQ.2 ) THEN
          WRITE(*,100) ' Enter Long. location of Vert. plane'
          READ(*,*) X
         WRITE(*,100) ' Enter init., final, and delta Lateral pos.'
          READ(*,*) ZINIT,ZFINL,DELTAZ
          WRITE(*,100) ' Enter init., final, and delta Height '
          WRITE(1,100) ' Lateral/Vertical plane at Longitudinal dist.'
          READ(*,*) YINIT,YFINL,DELTAY
          WRITE(1,11) X,VBAR,CT,GAMSIN,GAMCOS
11        FORMAT(1X,'  at, X:',F6.3,';  VBAR:',F5.3,';   CT:',
     >    F7.5,';    GAMSIN:',F5.3,';   GAMCOS:',F5.3)
          WRITE(1,100) '    X      Y      Z      DLAM      DMU       DNU
     >  GAMMA    R     PHI'
          WRITE(1,100) '--------- --------- --------- --------- --------- -
     >-------- --------- ---------'
          Z = ZINIT - DELTAZ
          Y = YINIT
C
C-----INCREMENT ALONG Z-DIRECTION (and X) for horizontal plane
C
       ELSE IF ( IDIR.EQ.3 ) THEN
          WRITE(*,100) ' Enter height of Horizontal plane'
          READ(*,*) Y
          WRITE(*,100) ' Enter init., final, and delta Horiz. distance'
          READ(*,*) XINIT,XFINL,DELTAX
          WRITE(*,100) ' Enter init., final, and delta Lateral distance'
          READ(*,*) ZINIT,ZFINL,DELTAZ
          WRITE(1,100) ' Horizontal plane at Vertical Height'
          WRITE(1,12) Y,VBAR,CT,GAMSIN,GAMCOS
12        FORMAT(1X,'   at, Y:',F6.3,';  VBAR:',F5.3,';   CT:',
     >    F7.5,';    GAMSIN:',F5.3,';   GAMCOS:',F5.3)
      WRITE(1,100) '    X      Y      Z      DLAM      DMU       DNU
     > GAMMA    R     PHI'
      WRITE(1,100) '--------- --------- --------- --------- --------- -
     >-------- --------- ---------'
          X = XINIT - DELTAX
```

```
          Z = ZINIT
C
C-----INCREMENT ALONG RADIAL - for various azimuths for a horiz. plane
C
       ELSE IF ( IDIR.EQ.4 ) THEN
          WRITE(*,100) ' Enter Height of horizontal plane'
          READ(*,*) Y
          WRITE(*,100) ' for Radial variation at various Azimuths'
          WRITE(*,100) ' Enter initial, final, and delta Radius'
          READ(*,*) RINIT,RFINL,DELTAR
          WRITE(*,100) ' Enter initial, final, and delta Azimuth'
          READ(*,*) AZINIT,AZFINL,DELTAP
          WRITE(1,100) ' Radial variation at various Azimuths'
          WRITE(1,13) Y,VBAR,CT,GAMSIN,GAMCOS
13        FORMAT(1X,' at Height, Y:',F6.3,'; VBAR:',F5.3,'; CT:',
     >   F7.5,'; GAMSIN:',F5.3,'; GAMCOS:',F5.3)
          WRITE(1,100) '    AZ      R      DLAM      DMU      DNU
     >   GAMMA'
          WRITE(1,100) '--------- --------- --------- --------- --------- -
     >--------- --------- ---------'
          RBAR = RINIT - DELTAR
          PHI = AZINIT
C
C-----INCREMENT ALONG ANNULUS for various radii for a horiz. plane
C
       ELSE IF (IDIR.EQ.5) THEN
          WRITE(*,100) ' Enter height of horizontal plane'
C
          READ(*,*) Y
          WRITE(*,100) ' for Azimuthal variation at various Radii'
          WRITE(*,100) ' Enter initial, final, and delta Azimuth'
          READ(*,*) AZINIT,AZFINL,DELTAP
          WRITE(*,100) ' Enter initial, final, and delta Radii'
          READ(*,*) RINIT,RFINL,DELTAR
          WRITE(1,100) ' Azimuthal variation at various Radii'
          WRITE(1,14) Y,VBAR,CT,GAMSIN,GAMCOS
14        FORMAT(1X,' at Height, Y:',F6.3,'; VBAR:',F5.3,'; CT:',
     >   F7.5,'; GAMSIN:',F5.3,'; GAMCOS:',F5.3)
          WRITE(1,100) '    AZ      R      DLAM      DMU      DNU
     >   GAMMA'
          WRITE(1,100) '--------- --------- --------- --------- --------- -
     >--------- --------- ---------'
          RBAR = RINIT
```

```
            PHI = AZINIT - DELTAP
            AZFINLX = AZFINL - DELTAP
        ENDIF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       LOOP POINT FOR INTEGRATION CALCULATIONS
C
        WRITE(*,'(1X,80A)') ' CALCULATING. . .'
15      CONTINUE
        IF ( (IDIR.EQ.1).AND.(X.GE.XFINL).AND.(Y.GE.YFINL) )
     >  GOTO 999
        IF ( (IDIR.EQ.1).AND.(X.GT.XFINL) ) THEN
            X = XINIT
            Y = Y + DELTAY
        ELSE IF (IDIR.EQ.1) THEN
            X = X + DELTAX
        ENDIF
C
        IF ( (IDIR.EQ.2).AND.(Z.GE.ZFINL).AND.(Y.GE.YFINL) )
     >  GOTO 999
        IF ( (IDIR.EQ.2).AND.(Z.GT.ZFINL) ) THEN
            Z = ZINIT
            Y = Y + DELTAY
        ELSE IF (IDIR.EQ.2) THEN
            Z = Z + DELTAZ
        ENDIF
C
        IF ( (IDIR.EQ.3).AND.(X.GE.XFINL).AND.(Z.GE.ZFINL) )
     >      GOTO 999
        IF ( (IDIR.EQ.3).AND.(X.GT.XFINL) ) THEN
            X = XINIT
            Z = Z + DELTAZ
        ELSE IF (IDIR.EQ.3) THEN
            X = X + DELTAX
        ENDIF
C
       IF ( (IDIR.EQ.4).AND.(RBAR.GE.RFINL).AND.(PHI.GE.AZFINL) )
     >  GOTO 999
        IF ( (IDIR.EQ.4).AND.(RBAR.GT.RFINL) ) THEN
            RBAR = RINIT
            PHI = PHI + DELTAP
        ELSE IF  (IDIR.EQ.4) THEN
            RBAR = RBAR + DELTAR
```

```
      ENDIF
C
      IF ( (IDIR.EQ.5).AND.(PHI.GE.AZFINL).AND.(RBAR.GE.RFINL) )
   >     GOTO 999
      IF ( (IDIR.EQ.5).AND.(PHI.GT.AZFINLX) ) THEN
         PHI = AZINIT
         RBAR = RBAR + DELTAR
      ELSE IF (IDIR.EQ.5) THEN
        PHI = PHI + DELTAP
      ENDIF
C
      IF (IDIR.LE.3) THEN
        RBAR = SQRT (X*X + Z*Z)
      IF (RBAR.LT.0.01) RBAR = .01
         SINAZ =  Z/RBAR
         COSAZ = -X/RBAR
         AZ = ASIN(SINAZ)
         PHI = AZ/D2R
      IF ((SINAZ.GE.0.).AND.(COSAZ.GE.0.)) PHI = PHI
      IF ((SINAZ.GE.0.).AND.(COSAZ.LT.0.)) PHI = 180. - PHI
      IF ((SINAZ.LT.0.).AND.(COSAZ.LT.0.)) PHI = 180. - PHI
      IF ((SINAZ.LT.0.).AND.(COSAZ.GE.0.)) PHI = 360. + PHI
        COS2AZ = COS(2*PHI*D2R)
        GAMFACT = 1. - GAMSIN*VBAR*SINAZ + GAMCOS*VBAR*VBAR*(1.
   > - COS2AZ)
      ENDIF
C
      IF ( IDIR.GE.4 ) THEN
         X = -RBAR*COS(PHI*D2R)
         Z =  RBAR*SIN(PHI*D2R)
         SINAZ = SIN(PHI*D2R)
         COS2AZ = COS(2*PHI*D2R)
         GAMFACT = 1. - GAMSIN*VBAR*SINAZ + GAMCOS*VBAR*VBAR*(1.
   > - COS2AZ)
      ENDIF
C
        VX = 0.0
        VY = 0.0
        VZ = 0.0
C
        DO 50 II = 1,NSEG
C
          RHO = II/FLOAT(NSEG)
```

```
      VBARSTAR = VBAR/RHO
      X1     = X/RHO
      XINF = -20.0/RHO
      Y1     = Y/RHO
      YINF = Y1
      Z1     = Z/RHO
      ZINF = Z1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     FIRST INTEGRAL AROUND DISK FROM PI/2 TO 3PI/2, PRODUCING
C     FACTORS K,L,M AND N.  BG AND ED ARE THE LIMITS OF INTEGRATION,
C     S IS INTEGRAND.
C
      BG = PI/2.
      ED = 3.*PI/2.
      S  = 0.01
      SINV2P = S / (2.0*PI)
      A = BG
      K =   0.0
      KINF = 0.0
      L =   0.0
      LINF = 0.0
      M =   0.0
      MINF = 0.0
      N =   0.0
      NINF = 0.0
      U =   0.0
      UINF = 0.0
      B1 = 0.0
      B1INF = 0.0
      B2 = 0.0
C
      B2INF = 0.0
      W1A = 0.0
      W1AINF = 0.0
      W2A = 0.0
      W2AINF = 0.0
      W1B = 0.0
      W1BINF = 0.0
      W2B = 0.0
      W2BINF = 0.0
      W1 = 0.0
      W1INF = 0.0
```

```
      W2 = 0.0
      W2INF = 0.0
C
20    CONTINUE
      IF (A.LT.ED) THEN
         CS1 = COS(A)
         CS2 = COS(A+S)
         SN1 = SIN(A)
         SN2 = SIN(A+S)
         B1 = Y1*Y1 + (SN1-Z1)*(SN1-Z1)
         B1INF = YINF*YINF + (SN1-ZINF)*(SN1-ZINF)
         B2 = Y1*Y1 + (SN2-Z1)*(SN2-Z1)
         B2INF = YINF*YINF + (SN2-ZINF)*(SN2-ZINF)
         W1A = (CS1-X1)/SQRT((CS1-X1)*(CS1-X1)+B1)
         W1AINF = (CS1-XINF)/SQRT((CS1-XINF)*(CS1-XINF)+B1INF)
         W1B = (CS1+X1)/SQRT((CS1+X1)*(CS1+X1)+B1)
         W1BINF = (CS1+XINF)/SQRT((CS1+XINF)*(CS1+XINF)+B1INF)
         W1 = W1A - W1B
         W1INF = W1AINF - W1BINF
         W2A = (CS2-X1)/SQRT((CS2-X1)*(CS2-X1)+B2)
         W2AINF = (CS2-XINF)/SQRT((CS2-XINF)*(CS2-XINF)+B2INF)
         W2B = (CS2+X1)/SQRT((CS2+X1)*(CS2+X1)+B2)
         W2BINF = (CS2+XINF)/SQRT((CS2+XINF)*(CS2+XINF)+B2INF)
         W2 = W2A - W2B
         W2INF = W2AINF - W2BINF
         K = K + (W1/B1) + (W2/B2)
         KINF = KINF + (W1INF/B1INF) + (W2INF/B2INF)
         L = L + (SN1*W1/B1) + (SN2*W2/B2)
         LINF = LINF + (SN1*W1INF/B1INF) + (SN2*W2INF/B2INF)
         U = U + (SN1*SN1*W1/B1) + (SN2*SN2*W2/B2)
         UINF = UINF + (SN1*SN1*W1INF/B1INF) + (SN2*SN2*W2INF/B2INF)
         A = A + S
      GO TO 20
      ENDIF
C
C   SWAP ORDER OF CALCULATION TO AVOID 0.0/0.0 SINGULARITY
C
      N = (U-Z1*L)*SINV2P
      NINF = (UINF-Z1*LINF)*SINV2P
      M = (L-Z1*K)*SINV2P
      MINF = (LINF-Z1*KINF)*SINV2P
C
      K = K*SINV2P*Y1
```

```
            KINF = KINF*SINV2P*YINF
            L = L*SINV2P*Y1
            LINF = LINF*SINV2P*YINF
            U = U*SINV2P
            UINF = UINF*SINV2P
C     (OLD METHOD USED UPDATED K,L,U VALUES)
C           N = U-(Z1*L/Y1)
C           NINF = UINF-(ZINF*LINF/YINF)
C           M = (L-Z1*K)/Y1
C           MINF = (LINF-ZINF*KINF)/YINF
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     SECOND INTEGRAL OVER LIMITS -1 TO 1 PRODUCES FACTORS H AND J.
C
            H   = 0.0
            H1  = 0.0
            H1A = 0.0
            H2B = 0.0
            H2  = 0.0
            H2A = 0.0
            H2B = 0.0
            YX  = 0.0
            YX11 = 0.0
            YX21 = 0.0
            YX31 = 0.0
            YX12 = 0.0
            YX22 = 0.0
            YX32 = 0.0
            BG = -1.0
            ED =  1.0
            A  =  BG
C
            YX11  = Y1/(Y1*Y1 + (A-Z1)**2)
            YX21A = X1 + SQRT(1-A*A)
            YX21B = SQRT((X1 + SQRT(1-A*A))**2 + Y1*Y1 + (A-Z1)**2)
            YX21  = YX21A/YX21B
            YX31A = X1 - SQRT(1-A*A)
            YX31B = SQRT((X1 - SQRT(1-A*A))**2 + Y1*Y1 + (A-Z1)**2)
            YX31  = YX31A/YX31B
            YX01  = YX11*(YX21 - YX31)
            H1A = 1./YX21B
            H1B = 1/YX31B
            H1  = H1A - H1B
```

```
30          CONTINUE
C
            IF (A.LE.(ED-S)) THEN
               A     = A + S
               YX12  = Y1/(Y1*Y1 + (A - Z1)**2)
               YX22A = X1 + SQRT(1-A*A)
               YX22B = SQRT((X1 + SQRT(1-A*A))**2 + Y1*Y1 + (A-Z1)**2)
               YX22  = YX22A/YX22B
               YX32A = X1 - SQRT(1-A*A)
               YX32B = SQRT((X1 - SQRT(1-A*A))**2 + Y1*Y1 + (A-Z1)**2)
               YX32  = YX32A/YX32B
               YX02    = YX12*(YX22 - YX32)
               YX     = YX + YX01 + YX02
               YX01 = YX02
               H2A = 1./YX22B
               H2B = 1./YX32B
               H2  = H2A - H2B
               H     = H + H1 + H2
               H1  = H2
            GOTO 30
            ENDIF
            H = H * (-1.) * 0.5 * SINV2P
            J = YX * (-1.) * SINV2P * 0.5
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    INTEGRAL FACTORS COMPLETE.  NOW BUILD INFLUENCE COEFF'S, AND
C    MULTIPLY BY CIRCULATION DISTRIBUTION TO GET VELOCITIES.
C
            DVYI(II) = H - 0.5*(VBARSTAR * (MINF + M) + (NINF + N))
            DVZI(II) = -0.5 * (VBARSTAR * (KINF + K) + (LINF + L))
            DVXI(II) = J
C
            VY = VY + DVYI(II) * DGAMMA(II) * GAMFACT
            VZ = VZ + DVZI(II) * DGAMMA(II) * GAMFACT
            VX = VX + DVXI(II) * DGAMMA(II) * GAMFACT
50    CONTINUE

C
C         NORMALIZE THE VELOCITY COMPONENTS AND PRINT OUTPUT
C
          VO = CT / (2.0*VBAR)
          LAMBDA = VY * VO
          MU = -1.0 * VX * VO
```

```
      NU = VZ * VO
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     UPDATE DESIRED EVALUATION POSITION, AND BRANCH BACK
C
      IF (IDIR.LE.3) THEN
        WRITE(1,56) X, Y, Z, LAMBDA, MU, NU, GAMFACT,RBAR,PHI
        WRITE(*,56) X, Y, Z, LAMBDA, MU, NU, GAMFACT,RBAR,PHI
56      FORMAT(1X,3(F6.2,1X),3(F10.5,1X),F8.4,F6.2,F8.0)
      ELSE
        WRITE(1,57) PHI, RBAR, LAMBDA, MU, NU, GAMFACT
        WRITE(*,57) PHI, RBAR, LAMBDA, MU, NU, GAMFACT
57      FORMAT(1X,F8.0,2X,F6.2,2X,3(F10.5,2X),F9.4)
      ENDIF
C
      GO TO 15
C
C     END OF ITERATIONS; CLOSE DATA FILE.
C
999   CONTINUE
      CLOSE(1)
      STOP
      END
```

APPENDIX C

CIRCULATION SUBROUTINES

```
C
C     GA1  FUNCTION FOR THE CIRCULATION DISTRIBUTION
C          Linear distribution (fig. 3.3 of ref. 1)
C
      REAL FUNCTION GA(A)
      REAL A
C
      GA = 10.0 * (0.0589 + 1.3783*A)
C
      RETURN
      END
```

```
C
C     GA2  FUNCTION FOR THE CIRCULATION DISTRIBUTION
C          Parabolic distribution (page 56 of ref. 1)
C
      REAL FUNCTION GA(A)
      REAL A
C
      GA = 10.0 * (A*A - A**3)
C
      RETURN
      END
```

```
C
C     GA3  FUNCTION FOR THE CIRCULATION DISTRIBUTION
C          EQ. 3.34 (of ref. 1) for blades with -10 deg twist
C
      REAL FUNCTION GA(A)
      REAL A
C
      GA = ((1.0 - 0.685*A)*A*(A*A + 0.011))/(A*A + 0.0055)
C
      RETURN
      END
```

APPENDIX D

In the code the circulation is normalized, so that any pattern can be coupled to the code, as follows:

$$G(\text{normal}) \equiv 2\int_0^1 \Gamma(\rho)\,\rho\, d\rho$$

$$\Gamma(\text{normal}) \equiv \frac{\Gamma(\rho)}{G} = \frac{\Gamma(\rho)}{2\int_0^1 \Gamma(\rho)\,\rho\, d\rho}$$

This results in velocity components normalized by the momentum value of the downwash independent of the choice of $\Gamma(\rho)$, where :

$$V_y = \frac{v_y}{v_0} = \frac{\lambda_i}{\overline{v}_0}$$

$$V_z = \frac{v_z}{v_0} = \frac{\eta_i}{\overline{v}_0}$$

$$V_x = \frac{v_x}{v_0} = \frac{-\mu_i}{\overline{v}_0}$$

and where $$\overline{v}_0 = \frac{v_0}{V_t} = \frac{C_T}{2\mu}$$

and then as final output

$$\Delta\lambda = \left(\frac{\lambda_i}{v_0}\right)\left(\frac{C_T}{2\mu}\right)$$

$$\Delta\eta = \left(\frac{\eta_i}{v_o}\right)\left(\frac{C_T}{2\mu}\right)$$

$$\Delta\mu = \left(\frac{-\mu_i}{v_o}\right)\left(\frac{C_T}{2\mu}\right)$$

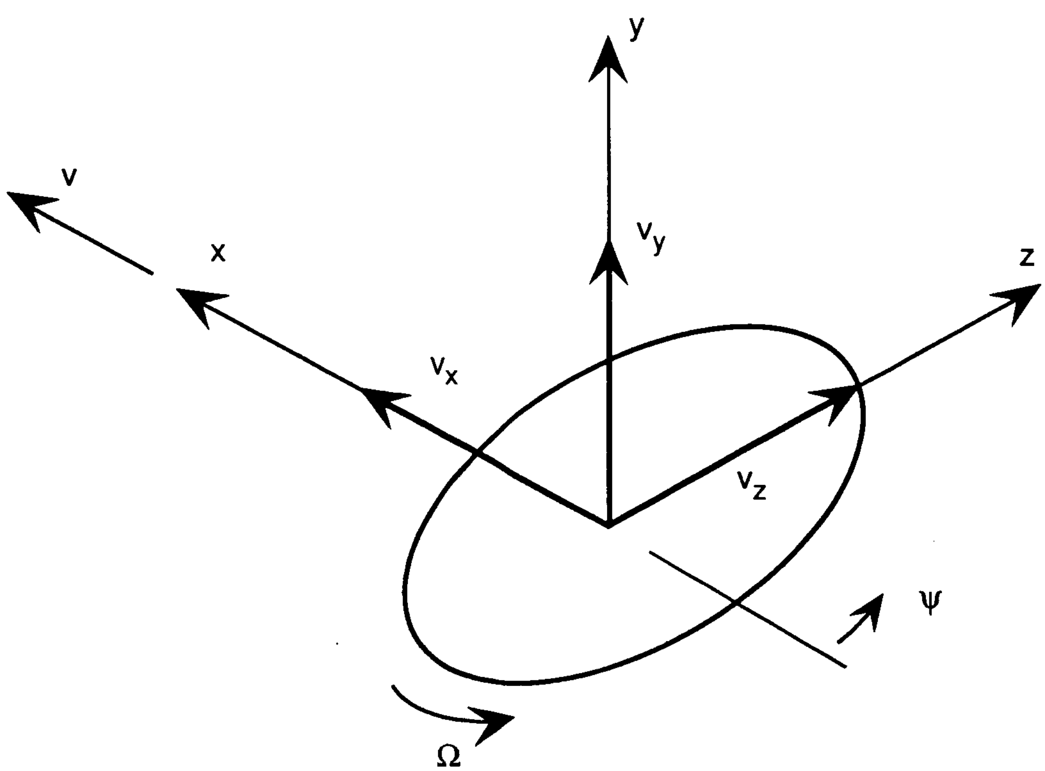

Figure 1. Coordinate System

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA TM-104139
AVSCOM TR-91-B-016

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
User's Guide for a "Flat Wake" Rotor Inflow/Wake Velocity Prediction Code, "DOWN"

5. Report Date
November 1991

6. Performing Organization Code

7. Author(s)
John C. Wilson

8. Performing Organization Report No.

10. Work Unit No.
505-59-36-01

9. Performing Organization Name and Address
Aerostructures Directorate
U.S. Army-AVSCOM
Langley Research Center
Hampton, VA 23665-5225

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001 and
U.S. Army Aviation Systems Command
St. Louis, MO 63120-1798

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer code named DOWN has been created to implement a "flat wake theory" for the calculation of rotor inflow and wake velocities. The theory was developed by V. E. Baskin of the USSR. The code was developed at Princeton University under a Grant from the National Aeronautics and Space Administration (NASA). A brief description of the code methodology and instructions for its use are given. The code will be available from NASA's Computer Software Management and Information Center (COSMIC).

17. Key Words (Suggested by Author(s))
Rotor inflow/wake
Code "DOWN"
Flat wake theory

18. Distribution Statement
[redacted]
Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	26	