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

John C. Wilson

November 1991

Review for general release November 30, 1993
The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products by the National Aeronautics and Space Administration.
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) \hspace{1cm} \text{rotor thrust coefficient,} \ T/[(\rho a (\Omega R)^2 \pi R^2)]

\[ G \] \hspace{1cm} \text{circulation "normalizing factor"}

\[ r \] \hspace{1cm} \text{rotor blade station, ft}

\[ R \] \hspace{1cm} \text{rotor radius, ft}

\[ T \] \hspace{1cm} \text{rotor thrust, lbs}

\[ V_o \] \hspace{1cm} \text{momentum theory value of average induced velocity, fps}

\[ \frac{V_o}{V_T} \] \hspace{1cm} \text{\( V_o \) normalized with tip speed i.e. \( \frac{V_o}{V_T} \)}

\[ V \] \hspace{1cm} \text{wind or flight speed, ft/sec}

\[ V_t \] \hspace{1cm} \text{rotor tip speed, \( \Omega R \), ft/sec}

\[ v_x \] \hspace{1cm} \text{increment in longitudinal velocity due to rotor inflow/wake, positive forward, ft/sec}

\[ v_y \] \hspace{1cm} \text{increment in vertical velocity due to rotor inflow/wake, positive upward, ft/sec}

\[ v_z \] \hspace{1cm} \text{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) \hspace{1cm} \text{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, \( /R \))}

\[ \Gamma \] (GAMMA) \hspace{1cm} \text{blade circulation, ft}^2/\text{sec}

\[ (GAMSIN) \] \hspace{1cm} \text{factor for azimuthal variation of circulation as a function of sine of azimuth angle}

\[ (GAMCOS) \] \hspace{1cm} \text{factor for azimuthal variation of circulation as a function of cosine of azimuth angle}

\[ (GAMFACT) \] \hspace{1cm} \text{ratio of azimuthally affected circulation to assumed "average" circulation}

\[ \Delta \lambda \] (DLAM) \hspace{1cm} \text{increment in rotor inflow ratio, \( \frac{v_y}{V_t} \)}

\[ \Delta \mu \] (DMU) \hspace{1cm} \text{increment in rotor advance ratio, \( \frac{v_x}{V_t} \)}

\[ \mu \] (VBAR) \hspace{1cm} \text{rotor advance ratio, \( \frac{V}{V_t} \)}

\[ \Delta \eta \] (DNU) \hspace{1cm} \text{increment in sidewash velocity component, \( \frac{v_z}{V_t} \)}

\[ \rho \] (RBAR) \hspace{1cm} \text{rotor blade station, \( r/R \)}

\[ \rho_a \] \hspace{1cm} \text{air density, slugs/ft}^3

\[ \psi \] (PSI) \hspace{1cm} \text{blade azimuth position, deg}

\[ \Omega \] \hspace{1cm} \text{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 (i.e. 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 fac-
tors, 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, \( \mu \).

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 longitudinal 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 Δλ, Δμ, and Δη 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 Δλ, Δμ, and Δη 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

<table>
<thead>
<tr>
<th>PSI</th>
<th>r/R</th>
<th>DLAM</th>
<th>DMU</th>
<th>DNU</th>
<th>GAMFACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>.20</td>
<td>-.00025</td>
<td>.01301</td>
<td>.02746</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.</td>
<td>.40</td>
<td>-.00197</td>
<td>.01757</td>
<td>.03128</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.</td>
<td>.60</td>
<td>-.00784</td>
<td>.01987</td>
<td>.03249</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.</td>
<td>.80</td>
<td>-.01658</td>
<td>.01896</td>
<td>.03256</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.</td>
<td>1.00</td>
<td>-.02665</td>
<td>.00966</td>
<td>.03121</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.</td>
<td>1.20</td>
<td>-.02191</td>
<td>.00168</td>
<td>.03003</td>
<td>1.0000</td>
</tr>
<tr>
<td>30.</td>
<td>.20</td>
<td>-.01812</td>
<td>.01172</td>
<td>.02903</td>
<td>.9007</td>
</tr>
<tr>
<td>30.</td>
<td>.40</td>
<td>-.03214</td>
<td>.01582</td>
<td>.02172</td>
<td>.9007</td>
</tr>
<tr>
<td>30.</td>
<td>.60</td>
<td>-.04297</td>
<td>.01789</td>
<td>.01444</td>
<td>.9007</td>
</tr>
<tr>
<td>30.</td>
<td>.80</td>
<td>-.05509</td>
<td>.01708</td>
<td>.00454</td>
<td>.9007</td>
</tr>
<tr>
<td>30.</td>
<td>1.00</td>
<td>-.06681</td>
<td>.00870</td>
<td>-.01249</td>
<td>.9007</td>
</tr>
<tr>
<td>30.</td>
<td>1.20</td>
<td>-.05932</td>
<td>.00151</td>
<td>-.03101</td>
<td>.9007</td>
</tr>
<tr>
<td>60.</td>
<td>.20</td>
<td>-.02333</td>
<td>.01098</td>
<td>.01562</td>
<td>.8437</td>
</tr>
<tr>
<td>60.</td>
<td>.40</td>
<td>-.03120</td>
<td>.01482</td>
<td>.00555</td>
<td>.8437</td>
</tr>
<tr>
<td>60.</td>
<td>.60</td>
<td>-.03757</td>
<td>.01676</td>
<td>-.00673</td>
<td>.8437</td>
</tr>
<tr>
<td>60.</td>
<td>.80</td>
<td>-.04084</td>
<td>.01600</td>
<td>-.02364</td>
<td>.8437</td>
</tr>
<tr>
<td>60.</td>
<td>1.00</td>
<td>-.03639</td>
<td>.00815</td>
<td>-.05727</td>
<td>.8437</td>
</tr>
<tr>
<td>60.</td>
<td>1.20</td>
<td>-.04101</td>
<td>.00142</td>
<td>-.04796</td>
<td>.8437</td>
</tr>
<tr>
<td>90.</td>
<td>.20</td>
<td>-.02165</td>
<td>.01075</td>
<td>.00815</td>
<td>.8262</td>
</tr>
<tr>
<td>90.</td>
<td>.40</td>
<td>-.02520</td>
<td>.01452</td>
<td>-.00254</td>
<td>.8262</td>
</tr>
<tr>
<td>90.</td>
<td>.60</td>
<td>-.02417</td>
<td>.01642</td>
<td>-.01465</td>
<td>.8262</td>
</tr>
<tr>
<td>90.</td>
<td>.80</td>
<td>-.01640</td>
<td>.01565</td>
<td>-.02870</td>
<td>.8262</td>
</tr>
<tr>
<td>90.</td>
<td>1.00</td>
<td>.01875</td>
<td>.00776</td>
<td>-.03536</td>
<td>.8262</td>
</tr>
<tr>
<td>90.</td>
<td>1.20</td>
<td>.01472</td>
<td>.00138</td>
<td>-.00496</td>
<td>.8262</td>
</tr>
<tr>
<td>120.</td>
<td>.20</td>
<td>-.01896</td>
<td>.01098</td>
<td>.00426</td>
<td>.8437</td>
</tr>
<tr>
<td>120.</td>
<td>.40</td>
<td>-.01900</td>
<td>.01482</td>
<td>-.00490</td>
<td>.8437</td>
</tr>
<tr>
<td>120.</td>
<td>.60</td>
<td>-.01400</td>
<td>.01676</td>
<td>-.01290</td>
<td>.8437</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>120.</td>
<td>.80</td>
<td>-.00341</td>
<td>.01600</td>
<td>-.01890</td>
<td>.8437</td>
</tr>
<tr>
<td>120.</td>
<td>1.00</td>
<td>.01448</td>
<td>.00815</td>
<td>-.01276</td>
<td>.8437</td>
</tr>
<tr>
<td>120.</td>
<td>1.20</td>
<td>.00844</td>
<td>.00142</td>
<td>-.00197</td>
<td>.8437</td>
</tr>
<tr>
<td>150.</td>
<td>.20</td>
<td>-.01670</td>
<td>.01172</td>
<td>.00263</td>
<td>.9007</td>
</tr>
<tr>
<td>150.</td>
<td>.40</td>
<td>-.01506</td>
<td>.01582</td>
<td>-.00395</td>
<td>.9007</td>
</tr>
<tr>
<td>150.</td>
<td>.60</td>
<td>-.00914</td>
<td>.01789</td>
<td>-.00817</td>
<td>.9007</td>
</tr>
<tr>
<td>150.</td>
<td>.80</td>
<td>.00015</td>
<td>.01708</td>
<td>-.01009</td>
<td>.9007</td>
</tr>
<tr>
<td>150.</td>
<td>1.00</td>
<td>.01154</td>
<td>.00870</td>
<td>-.00582</td>
<td>.9007</td>
</tr>
<tr>
<td>150.</td>
<td>1.20</td>
<td>.00662</td>
<td>.00151</td>
<td>-.00093</td>
<td>.9007</td>
</tr>
<tr>
<td>180.</td>
<td>.20</td>
<td>-.01521</td>
<td>.01301</td>
<td>.00236</td>
<td>1.0000</td>
</tr>
<tr>
<td>180.</td>
<td>.40</td>
<td>-.01349</td>
<td>.01757</td>
<td>-.00147</td>
<td>1.0000</td>
</tr>
<tr>
<td>180.</td>
<td>.60</td>
<td>-.00763</td>
<td>.01987</td>
<td>-.00268</td>
<td>1.0000</td>
</tr>
<tr>
<td>180.</td>
<td>.80</td>
<td>.00111</td>
<td>.01896</td>
<td>-.00274</td>
<td>1.0000</td>
</tr>
<tr>
<td>180.</td>
<td>1.00</td>
<td>.01119</td>
<td>.00966</td>
<td>-.00139</td>
<td>1.0000</td>
</tr>
<tr>
<td>180.</td>
<td>1.20</td>
<td>.00644</td>
<td>.00168</td>
<td>-.00022</td>
<td>1.0000</td>
</tr>
<tr>
<td>210.</td>
<td>.20</td>
<td>-.01408</td>
<td>.01463</td>
<td>.00274</td>
<td>1.1242</td>
</tr>
<tr>
<td>210.</td>
<td>.40</td>
<td>-.01391</td>
<td>.01975</td>
<td>.00180</td>
<td>1.1242</td>
</tr>
<tr>
<td>210.</td>
<td>.60</td>
<td>-.00859</td>
<td>.02233</td>
<td>.00392</td>
<td>1.1242</td>
</tr>
<tr>
<td>210.</td>
<td>.80</td>
<td>.00087</td>
<td>.02132</td>
<td>.00585</td>
<td>1.1242</td>
</tr>
<tr>
<td>210.</td>
<td>1.00</td>
<td>.01289</td>
<td>.01086</td>
<td>.00375</td>
<td>1.1242</td>
</tr>
<tr>
<td>210.</td>
<td>1.20</td>
<td>.00748</td>
<td>.00189</td>
<td>.00062</td>
<td>1.1242</td>
</tr>
<tr>
<td>240.</td>
<td>.20</td>
<td>-.01231</td>
<td>.01602</td>
<td>.00320</td>
<td>1.2309</td>
</tr>
<tr>
<td>240.</td>
<td>.40</td>
<td>-.01397</td>
<td>.02162</td>
<td>.00468</td>
<td>1.2309</td>
</tr>
<tr>
<td>240.</td>
<td>.60</td>
<td>-.01296</td>
<td>.02445</td>
<td>.01134</td>
<td>1.2309</td>
</tr>
<tr>
<td>240.</td>
<td>.80</td>
<td>-.00244</td>
<td>.02334</td>
<td>.01805</td>
<td>1.2309</td>
</tr>
<tr>
<td>240.</td>
<td>1.00</td>
<td>.01725</td>
<td>.01189</td>
<td>.01287</td>
<td>1.2309</td>
</tr>
<tr>
<td>240.</td>
<td>1.20</td>
<td>.01028</td>
<td>.00207</td>
<td>.00203</td>
<td>1.2309</td>
</tr>
<tr>
<td>270.</td>
<td>.20</td>
<td>-.00847</td>
<td>.01657</td>
<td>.00365</td>
<td>1.2732</td>
</tr>
<tr>
<td>270.</td>
<td>.40</td>
<td>-.01794</td>
<td>.02238</td>
<td>.00469</td>
<td>1.2732</td>
</tr>
<tr>
<td>270.</td>
<td>.60</td>
<td>-.02184</td>
<td>.02530</td>
<td>.01519</td>
<td>1.2732</td>
</tr>
<tr>
<td>270.</td>
<td>.80</td>
<td>-.01656</td>
<td>.02411</td>
<td>.03120</td>
<td>1.2732</td>
</tr>
<tr>
<td>270.</td>
<td>1.00</td>
<td>.02242</td>
<td>.01195</td>
<td>.04074</td>
<td>1.2732</td>
</tr>
<tr>
<td>270.</td>
<td>1.20</td>
<td>.01809</td>
<td>.00212</td>
<td>.00578</td>
<td>1.2732</td>
</tr>
<tr>
<td>300.</td>
<td>.20</td>
<td>-.00150</td>
<td>.01602</td>
<td>.00605</td>
<td>1.2309</td>
</tr>
<tr>
<td>300.</td>
<td>.40</td>
<td>-.01444</td>
<td>.02162</td>
<td>.00155</td>
<td>1.2309</td>
</tr>
<tr>
<td>300.</td>
<td>.60</td>
<td>-.02797</td>
<td>.02445</td>
<td>.00850</td>
<td>1.2309</td>
</tr>
<tr>
<td>300.</td>
<td>.80</td>
<td>-.03789</td>
<td>.02334</td>
<td>.02437</td>
<td>1.2309</td>
</tr>
<tr>
<td>300.</td>
<td>1.00</td>
<td>-.03830</td>
<td>.01189</td>
<td>.06067</td>
<td>1.2309</td>
</tr>
<tr>
<td>300.</td>
<td>1.20</td>
<td>.04573</td>
<td>.00207</td>
<td>.05271</td>
<td>1.2309</td>
</tr>
<tr>
<td>330.</td>
<td>.20</td>
<td>.00468</td>
<td>.01463</td>
<td>.01560</td>
<td>1.1242</td>
</tr>
<tr>
<td>330.</td>
<td>.40</td>
<td>-.00105</td>
<td>.01975</td>
<td>.00464</td>
<td>1.1242</td>
</tr>
<tr>
<td>330.</td>
<td>.60</td>
<td>-.01545</td>
<td>.02233</td>
<td>.00044</td>
<td>1.1242</td>
</tr>
<tr>
<td>330.</td>
<td>.80</td>
<td>-.03255</td>
<td>.02132</td>
<td>.00243</td>
<td>1.1242</td>
</tr>
</tbody>
</table>
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


APPENDIX A

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

\[ \Gamma(p, \mu, \psi) = \Gamma(p) + \Delta \Gamma(\mu, \psi) \]

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:

\[ \Delta \Gamma(\mu, \psi) \approx -(\text{GAMSIN})(\text{VBAR}) \sin(\psi) + (\text{GAMCOS})(\text{VBAR})(\text{VBAR})(1 - \cos(2*\psi)) \]

where GAMSIN = 0.0 or 1.5

and GAMCOS = 0.0 or 1.2

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

\[ \text{GAMFACT} = \left( \Gamma(p) + \Delta \Gamma(\mu, \psi) \right) / \Gamma(p) \]
APPENDIX B

PROGRAM DOWN

CALCULATION OF ROTOR DOWNWASH FIELD USING FLAT-WAKE THEORY.

PROGRAM ORIGINALLY WRITTEN BY M. HAGLUND, '91, PRINCETON UNIV.

MODIFIED BY R. MCKILLIP 7/90 FOR IBM-PC USAGE.

ADDITIONAL MODIFICATIONS BY J. WILSON 10/90 -- 2/91

NSEG IS NUMBER OF ROTOR SEGMENTS

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)

REAL GAMMA(NSEG),DGAMMA(NSEG),DVYI(NSEG),DVZI(NSEG),DVXI(NSEG)
CHARACTER*10 FNAME
CHARACTER*10 CIRC

DATA PI/3.141592654/

OPEN FILES FOR OUTPUT

UNIT 1 IS DATA; PARAMETERS ARE PRINTED ON TITLE LINE.

WRITE(*,100) ' Enter filename for output data:'
READ(*,'(A)') FNAME
OPEN(1,FILE=FNAME)

WRITE(1,8) FNAME
8 FORMAT(1X,' Output File, FNAME ',10A)
WRITE(*,100) ' Enter Circulation Program used (i.e. GAL, etc.) '
READ(*,'(A)') CIRC
WRITE(1,9) CIRC
9 FORMAT(1X,' Circulation program ',10A)
CIRCULATION GAMMA IS COMPUTED USING A FUNCTION OF THE FORM:
GAMMA = GA(RHO), WHICH IS COMPILED SEPARATELY AND LINKED TO
THIS CODE, AS APPROPRIATE FOR THE TARGET COMPUTER.

THIS SECTION COMPUTES BOTH THE CIRCULATION GAMMA(I) AND THE
INCREMENTAL CIRCULATION DGAMMA(I).

FIRST, CALCULATE THE NORMALIZATION FACTOR

G = 0.0
NSEPS = 200
BG = 0.0
ED = 1.0
DRHO = (ED-BG)/FLOAT(NSTEPS)
RHO = BG
RHO2 = RHO + DRHO
GAMA = GA(RHO)

CONTINUE
IF (RHO2.LT.ED) THEN
    GAMMAD = GA(RHO2)
    G = G + (GAMA*RHO) + (GAMMAD*RHO2)
    GAMA = GAMMAD
    RHO = RHO2
    RHO2 = RHO2 + DRHO
    GOTO 4
ENDIF
G = G * DRHO

NOW CALCULATE THE CIRCULATION AND NORMALIZE BY G

DO 5 I=1,NSEG
    RHO = .5*(2*I-1)/FLOAT(NSEG)
    GAMMA(I) = GA(RHO)/G

CONTINUE
GAMMA(1) = 0.0

THIS LINE HAS BEEN REPLACED BY THE DO LOOP THATFollows.
GAMMA(11) = 0.0

...AND CONVERT NORMALIZED CIRCULATION TO A STAIR-STEP
FUNCTION FOR THE CALCULATION.
DO 6 I = 1, NSEG-1
DGAMMA(I) = GAMMA(I) - GAMMA(I+1)
6 CONTINUE
DGAMMA(NSEG) = GAMMA(NSEG)

CONTINUE

ASK FOR LOCATION AND TYPE OF VELOCITY CALCULATION

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) ?: '

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

WRITE(*,100) ' Enter ADVANCE RATIO: '
READ(*,*) VBAR

WRITE(*,100) ' Enter ROTOR THRUST COEFFICIENT: '
READ(*,*) CT

-----INCREMENT ALONG X-DIRECTION (and Y) for long./vert. plane

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(1,10) 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(1,10) 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
13
Z = ZINIT

C-----INCREMENT ALONG RADIAL - for various azimuths for a horiz. plane

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
    FORMAT(IX,' at Height, Y:',F6.3,'; VBAR:',F5.3,'; CT:',F7.5,'; GAMSIN:',F5.3,'; GAMCOS:',F5.3)
    > GAMMA'
    WRITE(1,100) '---- -------------- -------- -------- --------'
    >------------- -------------- -------- -------- --------
    RBAR = RINIT - DELTAR
    PHI = AZINIT

C-----INCREMENT ALONG ANNULUS for various radii for a horiz. plane

ELSE IF (IDIR.EQ.5) THEN
    WRITE(*,100) 'Enter height of horizontal plane'
    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
    FORMAT(IX,' at Height, Y:',F6.3,'; VBAR:',F5.3,'; CT:',F7.5,'; GAMSIN:',F5.3,'; GAMCOS:',F5.3)
    > GAMMA'
    WRITE(1,100) '------------ -------------- -------- -------- --------'
    >------------- -------------- -------- -------- --------
    RBAR = RINIT
\[
\begin{align*}
\Phi &= AZINIT - \text{DELTAP} \\
AZFINLX &= AZFINL - \text{DELTAP} \\
\text{ENDIF}
\end{align*}
\]

15 CONTINUE
LOOP POINT FOR INTEGRATION CALCULATIONS
WRITE(*,*(1X,80A)) 'CALCULATING...

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

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

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

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

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

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 = (CS2+X1)/SQRT((CS2+X1)*(CS2+X1)+B2)
    W1BINF = (CS2+XINF)/SQRT((CS2+XINF)*(CS2+XINF)+B2INF)
    WI = W1A - W1B
    WIINF = WI - WIINF
    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

SWAP ORDER OF CALCULATION TO AVOID 0.0/0.0 SINGULARITY

N = (U-Z1*L)*SINV2P
NINF = (UINF-Z1*LINF)*SINV2P
M = (L-Z1*K)*SINV2P
MINF = (LINF-Z1*KINF)*SINV2P

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)
NINF = UINF-(ZINF*LINF/YINF)
M = (L-Z1*K)/Y1
MINF = (LINF-ZINF*KINF)/YINF

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

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 NORMALIZE THE VELOCITY COMPONENTS AND PRINT OUTPUT

C VO = CT / (2.0*VBAR)
LAMBD4 = VY * VO
MU = -1.0 * VX * VO

20
NU = VZ * VO

UPDATE DESIRED EVALUATION POSITION, AND BRANCH BACK

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
ELSE
  WRITE(1,57) PHI, RBAR, LAMBDA, MU, NU, GAMFACT
  WRITE(*,57) PHI, RBAR, LAMBDA, MU, NU, GAMFACT
ENDIF

END

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

GA = 10.0 * (0.0589 + 1.3783*A)
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

GA = 10.0 * (A*A - A**3)
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

GA = ((1.0 - 0.685*A)*A*(A + 0.011))/(A*A + 0.0055)
RETURN
END
APPENDIX D

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

\[
G(\text{normal}) = \frac{1}{G} \int_0^1 \Gamma(\rho) \rho \, d\rho
\]

\[
\Gamma(\text{normal}) = \frac{\Gamma_{\text{normal}}}{G} = \frac{2}{\int_0^1 \Gamma(\rho) \rho \, d\rho}
\]

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

\[
V_y = \frac{v_y}{v_0} = \frac{\lambda_i}{\bar{v}_0}
\]

\[
V_z = \frac{v_z}{v_0} = \frac{\eta_i}{\bar{v}_0}
\]

\[
V_x = \frac{v_x}{v_0} = \frac{-\mu_i}{\bar{v}_0}
\]

and where

\[
\bar{v}_0 = \frac{v_0}{V_t} = \frac{C_T}{2\mu}
\]

and then as final output

\[
\Delta \lambda = \begin{pmatrix}
\lambda_i \\
\cdots \\
v_0
\end{pmatrix}
\begin{pmatrix}
C_T \\
\cdots \\
2\mu
\end{pmatrix}
\]

23
\[ \Delta \eta = \begin{pmatrix} \eta_i \\ v_0 \end{pmatrix} \begin{pmatrix} C_T \\ 2\mu \end{pmatrix} \]

\[ \Delta \mu = \begin{pmatrix} -\mu_i \\ v_0 \end{pmatrix} \begin{pmatrix} C_T \\ 2\mu \end{pmatrix} \]
Figure 1. Coordinate System
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).