Ryerson University
Aerospace Engineering

VAPTOR Manual

Author:
Devin Barcelos

Ryerson Applied Aerodynamics Laboratory of Flight

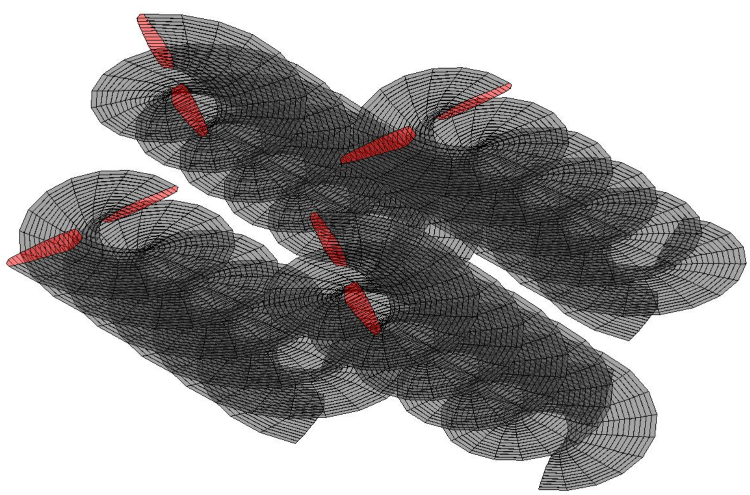
Last Updated: October 15, 2018

Contents

1 Overview

VAPTOR is a higher-order potential flow program created for performance predictions of rotors. It is used to model the aerodynamic forces and interactions of single rotors and/or multiple closely located rotors. Some of the key features include:

- Can model single rotors and multirotors
- Input basic information of rotor geometry and flow properties
- Setup for advance ratio and angle of attack performance sweeps
- Includes parallel computing features
- A timestepping procedure in which forces and wakes are computed
- A relaxed or prescribed wake can be simulated
- Filament free elements of distributed vorticity
- Models circulation as a second-order spline
- Viscous correction and stall model may be used
- Time-averaged loads, cyclic loads and radial loads can be extracted



2 Running Code

2.1 General Procedure

To begin running VAPTOR open MATLAB 2015a or newer and change the working folder to the VAPTOR folder. Within this directory there is a list of .m files where those beginning with 'fcn' are functions and VAPTOR_MAIN.m is the main code to be run. There are 2 folders: 'inputs' which contains all rotor/vehicle input files and 'airfoils' which contains the 2D viscous information of various airfoils.

The general procedure for running VAPTOR is:

1. create an input file if not already created
2. open VAPTOR_MAIN.m
3. within VAPTOR_MAIN.m modify strFILE to call desired input file
4. within VAPTOR_MAIN.m modify the flags and filename to be saved
5. run VAPTOR_MAIN.m
6. view results in the OUTPUT structure

Each of theses steps are described in more detail in the proceeding subsections. Parallel computing can be used for angle of attack and advance ratio sweeps using the VAPTOR_MAIN_PARALLEL.m using the same procedure.

2.2 Input File

2.2.1 VAPTOR Input File Description

The input file is used to define the blade geometry, the flow specifications and the simulation properties. This section will describe each input variable in more detail. Note: refer to the next section for an example input file.

Flow properties and simulation setup

flagRELAXWAKE	Toggles between a fixed wake analysis (0) and a relaxed wake analysis (1). Fixed wake prescribes the wake based on the inflow velocity and relax wake modifies the wake elements based on their experienced induced velocities. In general, a relaxed wake simulation requires about 10 times the computation time (and significantly more memory) than a fixed wake simulation. Relaxed wake is best for hover and descending flights.
flagSTEADY	Toggles between steady (1) and unsteady (2) analysis. It is strongly recommended to always use steady (1) analysis.

valMAXTIME	The maximum number of timestep iterations to be done. For rotor analysis this tends to be between 120 to 220 to reach convergence. The more edgewise flow present tends to require a higher valMAXTIME.
valAZNUM	The number of azimuth locations to be considered per rotation and used to adjust the timestep sizing. Often 20 azimuth locations are used but special cases (such as hover) require larger values of around 40.
seqALPHAR	The angle of attack definition. If interested in doing an angle of attack sweep, input multiple values for this. See section 4.2 for description angle of attack convention.
seqJ	The advance ratio definition. If interesting in doing an advance ratio sweep, input multiple values for this. See section 4.2 for description advance ratio convention.
valDENSITY	Air density to be use in viscous calculations.
valKINV	Air viscosity to be use in viscous calculations.
valDIA	The reference rotor diameter.
valNUMB	The number of blades per rotor.
valNUMRO	The total number of rotors

Rotor setup

Each of these variables must be defined for each rotor (see example input file).

matROTAX	Location (x,y,z) of the rotor rotational axis in the global reference frame.
vecRODIR	Rotational direction of the rotor where -1 is counterclockwise and 1 is clockwise.
valRPM	Rotational speed of rotor in revolutions per section.

Panel Geometries

Each of these variables except valPANELS and vecROTAXLOC must be defined for each panel. Note that a blade can be made up of several panels, each of which can have any number of chordwise or spanwise elements. It is important that all the elements on the same blade should have the same number of chordwise elements (vecM).

valPANELS	The number of panel per blade. Note that each panel can have multiple elements in both the chordwise and radial directions.
vecROTAXLOC	This is the local rotation axis for blade as defined by the following panel geometry.
vecN	Number of spanwise elements for the specific panel.
vecM	Number of chordwise elements for the specific panel.
vecAIRFOIL	The airfoil number that is associated to the desired file in the airfoil folder.
x,y,z	The coordinates of the left and right vertices of the panel leading edge. The first row of numbers is associated to the left vertex and the second row is associated to the right vertex.
chord	The chord length of the left and right sides of the specific panel as the first and second rows of numbers respectively.
beta	The blade pitch (in degrees) of the left and right sides of the specific panel as the first and second rows of numbers respectively.

If the rotor geometry is in a format that contains r/R, c/R, beta and mid-chordline, the GeometryCreation.m file in the geometry folder can be used to convert this format to VAPTOR format. To use GeometryGreation.m one must create a .mat table that contains the geometric information (see sample files as an example). In the load input data section specify the input .mat file and the rotor radius in meters. Run the script and copy the contents from the command window to the VAPTOR input file.

2.2.2 Example VAPTOR Input File

The following is an example input file for a quadrotor using four T-Motor 18x6.1 rotor. It is saved as a text file (.txt) and located in the inputs folder.

Input file for VAPTOR
Input file in m/N/sec
Please note that the program uses equal, number and : signs as special recognizers

Relaxed wake (yes 1, no 0):	flagRELAX = 0
Steady (1) or unsteady (2):	flagSTEADY = 1
Max. number of time steps:	valMAXTIME = 160
Number of azmith locations:	valAZNUM = 40
AOA sequence:	seqALPHAR = 5 15 30
Advance ratio:	seqJ = 0.4 0.6 0.8
Density:	valDENSITY = 1.2250
Kinematic viscosity:	valKINV = 1.460000e-05

Reference diameter:	valDIA = 0.4572
Number of blades:	valNUMB = 2
No. of rotors:	valNUMRO = 4
Rotor # 1	
Rotation axis (x y z):	matROTAX = 0.2828 0.2828 0.0
Rotation dir (CCW -1,CW 1):	vecRODIR = 1
Rotor RPM:	valRPM = 3000
Rotor # 2	
Rotation axis (x y z):	matROTAX = 0.2828 -0.2828 0.0
Rotation dir (CCW -1,CW 1):	vecRODIR = -1
Rotor RPM:	valRPM = 3000
Rotor # 3	
Rotation axis (x y z):	matROTAX = -0.2828 -0.2828 0.0
Rotation dir (CCW -1,CW 1):	vecRODIR = 1
Rotor RPM:	valRPM = 3000
Rotor # 4	
Rotation axis (x y z):	matROTAX = -0.2828 0.2828 0.0
Rotation dir (CCW -1,CW 1):	vecRODIR = -1
Rotor RPM:	valRPM = 3000
No. of panels:	valPANELS = 17

Defines leading edge of blades, measured in meters:
Keep vecM the same for all panels on a blade!

Local rotation axis	vecROTAXLOC = 0 0 0
Panel # :1	
Number of spanwise elements:	vecN = 1
Number of chordwise elements:	vecM = 1
Airfoil number:	vecAIRFOIL = 1

x	y	z	chord	pitch
-0.011244	0.034290	0.000000	0.025314	16.570000
-0.013329	0.045720	0.000000	0.033232	24.199900

⋮

Panel # :17	
Number of spanwise elements:	vecN = 1
Number of chordwise elements:	vecM = 1
Airfoil number:	vecAIRFOIL = 3

x	y	z	chord	pitch
-0.007915	0.217170	0.000000	0.021075	7.517000

0.002194	0.228600	0.000000	0.009098	7.526000

2.3 Airfoil Input File

In order to apply the viscous correction model an airfoil database must be create and saved to a .dat file. The airfoil database contains 5 columns and is often collected from simulations (such as xfoil) or experiment. The database contains 5 columns: angle of attack, lift coefficient, profile drag coefficient, Reynolds number and pitching moment coefficient. The first row is a title row and can have anything written for reference, such as the airfoil name.

It is suggested to do a large sweep of angles of attack and Reynolds numbers to incorporate the full range experienced by the rotor sections. The larger the database, the better. NOTE: the filename must be airfoil#.dat where # is the number that is referenced in the input file.

2.4 VAPTOR MAIN Inputs

Within the VAPTOR_MAIN.m file there are a few required inputs. These are located in the first section, titled '%% User Inputs'. This includes the input filename and a series of flags to toggle different VAPTOR routines.

strFILE	REQUIRED: A string to call the desired input file. Such as 'inputs/TMotorQuad.txt'
flagVISCOUS	This toggles the viscous correction model. This is a strip theory based method that uses the airfoil database as each panel location. NOTES: When on, viscous corrections will ONLY be applied to the last full rotation. This is done to save computation.
flagPRINT	This will print the results into the command window as the code progresses.
flagPLOT	This will create a wake plot of the rotor after the simulation is complete.
flagVERBOSE	This will add element numbers to the wake plot.
flagPROGRESS	This will create a progress bar which includes angle of attack, advance ratio and timestep.
flagHOVERWAKE	This will propagate the wake downward during a hover case. This propagation decreases linear and completes the simulation in a hover. It is important that valAZNUM < valMAXTIME for this. This is often used for lightly load rotors with relax wake.

flagSAVE	Save the full workspace to the provided filename. When not using parallel computing, this will save each timestep.
filename	The name of the file that the workspace should be save to.

2.5 Output File

In VAPTOR, a structure name 'OUTPUT' contains the forces and moment results at each timestep for each rotor, advance ratio and angle of attack. The force coefficients CT, CP, CFx, CFy, CQ, CMx, and CMy are the time-averaged results. The variables names matCT-TIMESTEP, matCPTIMESTEP etc. are the coefficients calculated at each timestep without time-averaging. These are all matrices with 4 dimension in the following order: timestep, rotor, advance ratio, angle of attack.

To extract data from this structure for a given case you can index each of these variables. For example, if the value of interest is the resulting thrust coefficient at an advance ratio of 0.8 and an angle of attack of 30°, one can use the following:

$$OUTPUT.CT(end,\ :,\ OUTPUT.seqJ == 0.8,\ OUTPUT.seqALPHAR == 30)$$

The first index is 'end' which is associated the time-averaged results of the last timestep, the second index is ':' referring to output all of the individual rotor results (for ex. rotors 1 to 4 in a quadcopter), the third index is the advance ratio and the fourth is the angle of attack index.

Note that if more variables are of interest (such as cyclic loading) they can either be simply added to the OUTPUT structure or taken from the saved workspace. If added to OUTPUT structure, it is suggested to preallocate the variable.

3 Code Structure

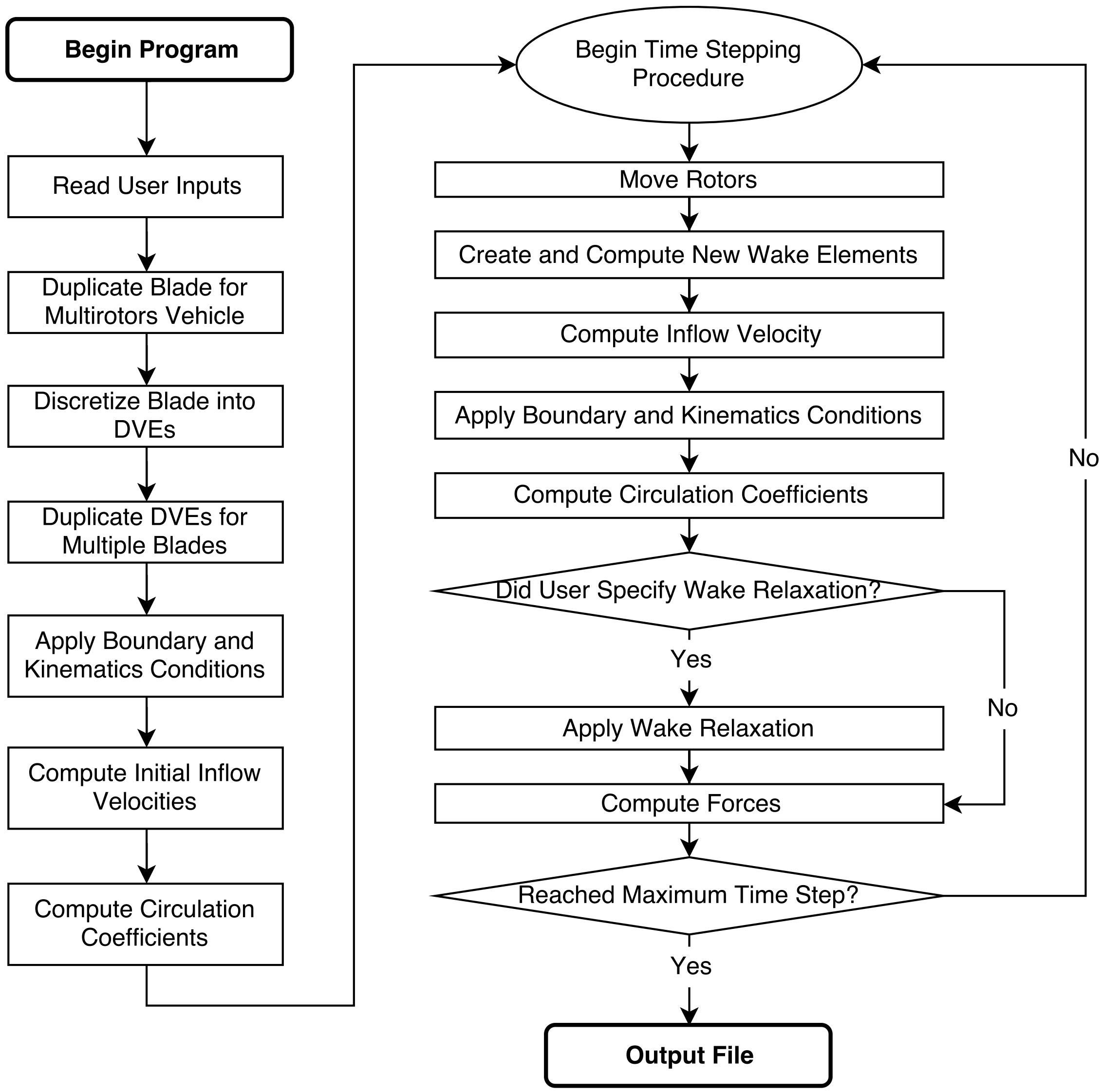


Figure 1: Program Execution Procedure

The general execution procedure is shown in Fig. 1. The program begins by reading the user created input file as explained in section 2.2. The geometry of a single blade is duplicated for the multirotor vehicle and this blade geometry is discretized into distributed vorticity elements (DVEs). The elements of the single blade are duplicated and rotated for each blade within each rotor. Initial boundary and kinematic conditions are computed and used in order to solve for the circulation of the lifting surface DVEs. The initial inflow velocity is calculated at each control point and along the trailing edge of the blade. The circulation is computed for the initial rotor.

The time stepping procedure is next initiated. This time stepping begins with moving the rotor and creating the new row of wake elements. The inflow velocity at the given time step is calculated to compute the circulation of the surface DVE. The wake relaxation process is applied depending on the user input specifications. The lift and drag forces acting on the rotors are computed and projected into rotor force conventions (thrust, torque, etc.). If the time stepping procedure is on the last full rotation of the rotor, both the viscous addition and the stall model are applied. The time stepping procedure continues to the next time step and increases until the user specified maximum number of time steps is reached. Once the maximum time step is reached, the output variables are saved. The full timestepping procedure is done for each angle of attack and advance ratio combination specified by the user.

4 Reference Notation

4.1 VAPTOR Variables

Within VAPTOR there are a few key notations that are used throughout the program. The variable names all begin with the variable type in lowercase followed by the variable description in uppercase. For example valMAXTIME is a single value (val) equal to the maximum number of timesteps (MAXTIME). The variable types include:

flag	Flag toggle
fcn	Function
mat	Matrix
seq	Number sequence
str	String
val	Single value
vec	Vector

It should be noted that a rotor in forward flight is simulated by moving in the negative x-direction and the positive z-direction as seen in Fig. 2.

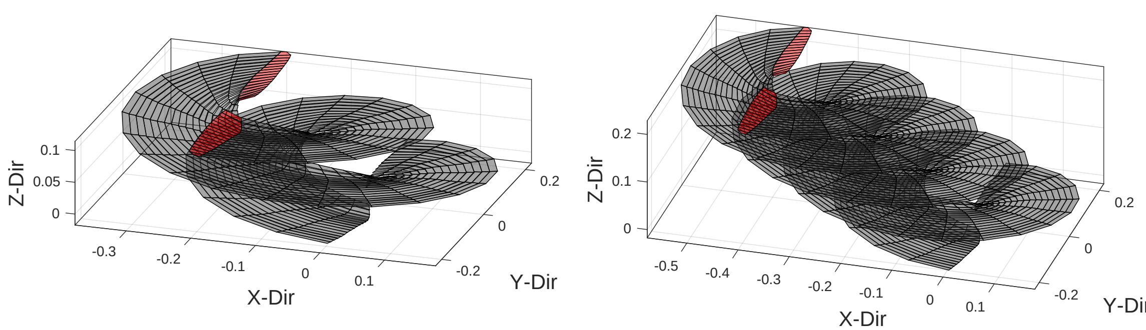


Figure 2: Single Rotor Progressing in x,y,z Axis

4.2 Rotor Notation

Figure 3 shows the angle of attack (α) convention used in VAPTOR. Note that 90° represents a vertical ascent in a quadcopter, 0° represents fully edgewise flow and $-90°$ represents vertical descent.

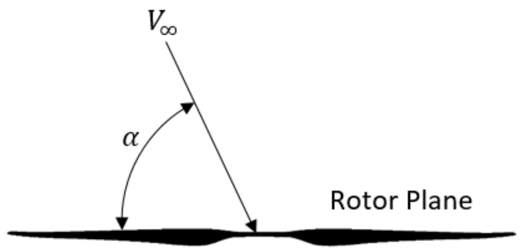


Figure 3: Angle of attack

Figure 4 shows the directions of the forces and moments acting on a rotor.

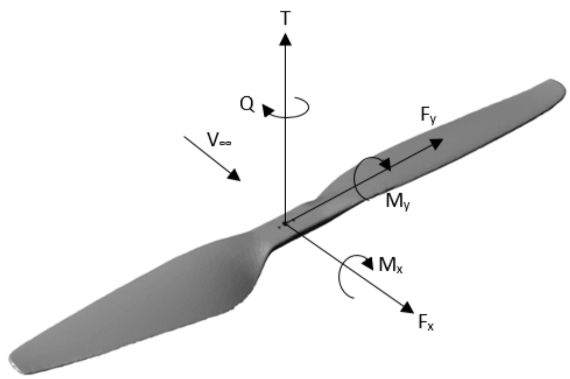


Figure 4: Forces and Moment Directions

The associated coefficients in VAPTOR are:

Thrust	$C_T = \frac{T}{\rho n^2 D^4}$
Torque	$C_Q = \frac{Q}{\rho n^2 D^5}$
Power	$C_P = \frac{P}{\rho n^3 D^5}$
Longitudinal Force	$C_{F_x} = \frac{F_x}{\rho n^2 D^4}$
Lateral Side Force	$C_{F_y} = \frac{F_y}{\rho n^2 D^4}$
Rolling Moment	$C_{M_x} = \frac{M_x}{\rho n^2 D^5}$
Pitching Moment	$C_{M_y} = \frac{M_y}{\rho n^2 D^5}$

Advance ratio (J) is defined as:

$$J = \frac{V}{nD}$$

Where ρ, n, D and V represent density, rotor rpm, rotor diameter and freestream velocity respectively.