UNIVERSITY OF
Southampton
School of Engineering Sciences

Aerodynamics & Flight Mechanics Research Group

Blade Vortex Intersection - A Geometrical Examination

S. J. Newman

Technical Report AFM-11/01

January 2011

UNIVERSITY OF SOUTHAMPTON

SCHOOL OF ENGINEERING SCIENCES

AERODYNAMICS AND FLIGHT MECHANICS RESEARCH GROUP

Blade Vortex Intersection - A Geometrical Examination

by

S. J. Newman

AFM Report No. AFM 11/01

January 2011

Introduction

The motion of a helicopter rotor in forward flight is a combination or rotation and translation and the combination of these two separate motions defines the path taken by any point of a rotor blade. This information is fundamental to the positioning of the vortex wake produced by the rotor.

Motion of a Rotor in Forward Flight

Plate 1 – Cobra AH-1W Descending to USS Kearsage

In order to define a wake position, the interrelationship between the rotor translation and rotation must be established. The gearing between these two components is the ratio of the forward speed (V) to the rotational speed of the rotor (Ω).

In order to determine this gearing, consider the translation and rotation over a time of Δt; these are as follows:

$$\begin{aligned} Forward\ Displacement &= V \cdot \Delta t \\ Rotation\,(\psi) &= \Omega \cdot \Delta t \end{aligned} \tag{1.}$$

If, instead of time t we use the rotational angle (ψ) as the independent variable, we find that these become:

$$\begin{aligned} Forward\ Displacement &= \frac{V}{\Omega} \cdot \psi \\ Rotation\,(\psi) &= \psi \end{aligned} \tag{2.}$$

Finally, if we scale the rotor to unit radius we find the scaled displacements to be:

$$\begin{aligned} \frac{Forward\ Displacement}{R} &= \frac{V}{\Omega R} \cdot \psi \\ &= \mu \cdot \psi \\ Rotation\,(\psi) &= \psi \end{aligned} \tag{3.}$$

From this analysis, the link between rotation and translation can be seen to be the advance ratio (μ).

This can be viewed as shown in Figure 1:

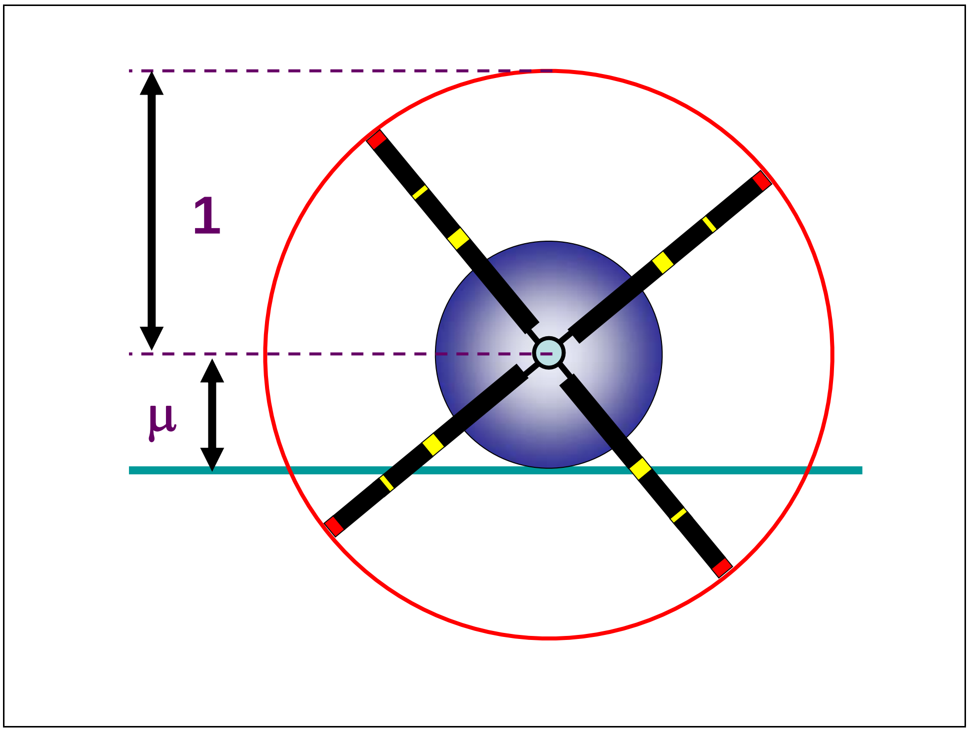

Figure 1 - Geometry of the Motion of a Rotor in Forward Flight

In this illustration, the rotor is of unit radius and can be regarded as being fixed to a circular drum of radius equal to the advance ratio. As this drum rolls along a straight edge, aligned to the forward flight direction, the rotor describes the correct movement for that advance ratio. The definition of a cycloid is the locus of a point of a circle whilst rolling - without slipping – along a straight line. Reference to Figure 1 shows the rotor performing exactly this situation which forms the origin of the cycloidal nature of the blade tip paths.

Generation of Blade Tip Vortices

The tip vortex emanating from a helicopter rotor blade tip can interact with a blade from the same rotor. Plate 1 shows a vortex wake from the main rotor of a Cobra AH-1W aircraft in descent before landing on a ship. The interaction is complex but the geometrical topology of the vortex wake can be viewed effectively by assuming that the vortex does not move relative to the air once it has been generated. The vortex trails will now be defined by the path taken by each rotor blade tip and as has been previously discussed is of a cycloidal profile in plan.

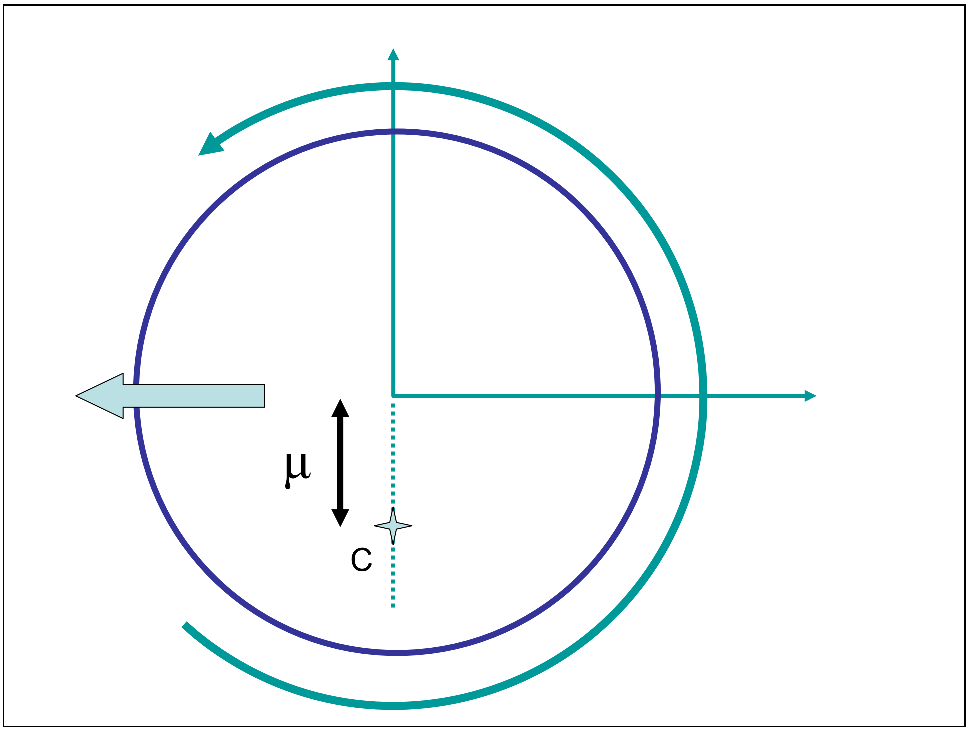

Figure 2 – Motion of Rotor and Instantaneous Centre of Rotation

Figure 2 shows a schematic diagram of the rotor movement. The rotor is assumed to be translating from right to left and rotating in an anticlockwise sense. *(This is commensurate with UK and USA helicopters).* As already discussed, the rotor is scaled to have unit radius. The rotor possesses an instantaneous centre of rotation (C) as shown in Figure 2. The location of C is on the retreating half of the disc with blade azimuth of 270° and displaced from the rotor centre by μ, the advance ratio.

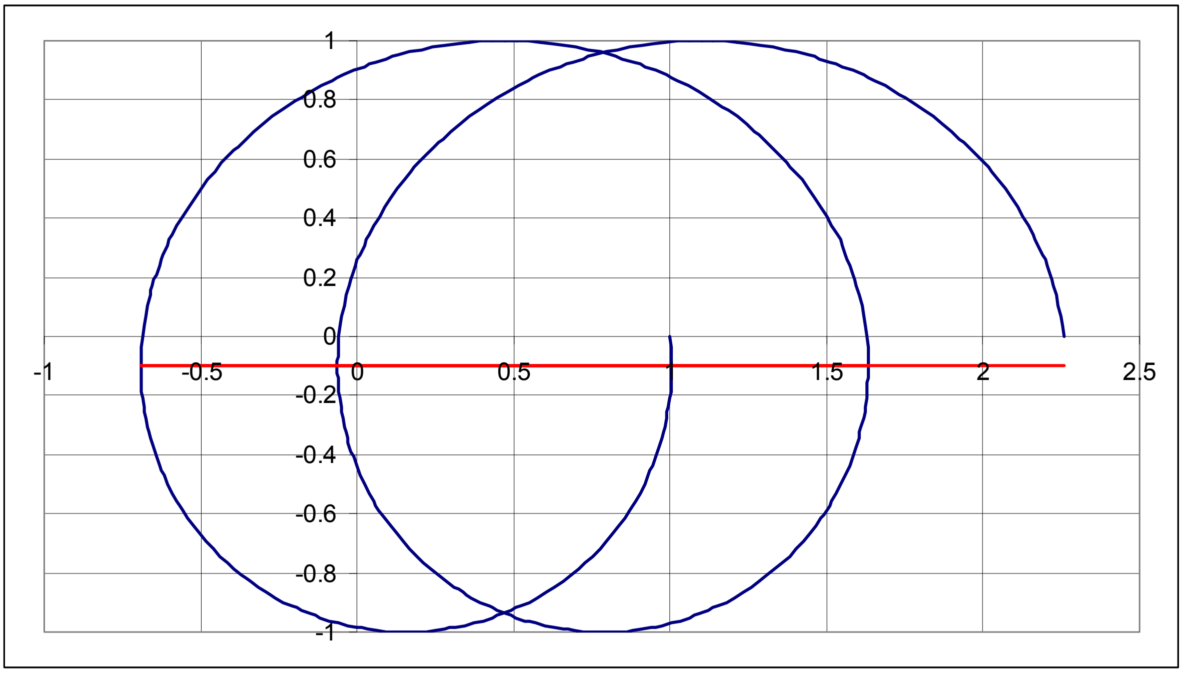

Figure 3 – Typical Cycloidal Vortex Trail

Figure 3 shows a typical curve generated by a tip path creating a vortex trail. It is that of a prolate cycloid. The shape of the cycloid is governed by the advance ratio, which is 0.1 for Figure 3. The line represents the locus of the instantaneous centre of rotation. A point to note is that the cycloidal vortex trajectory is at right angles to this locus at the points of intersection.

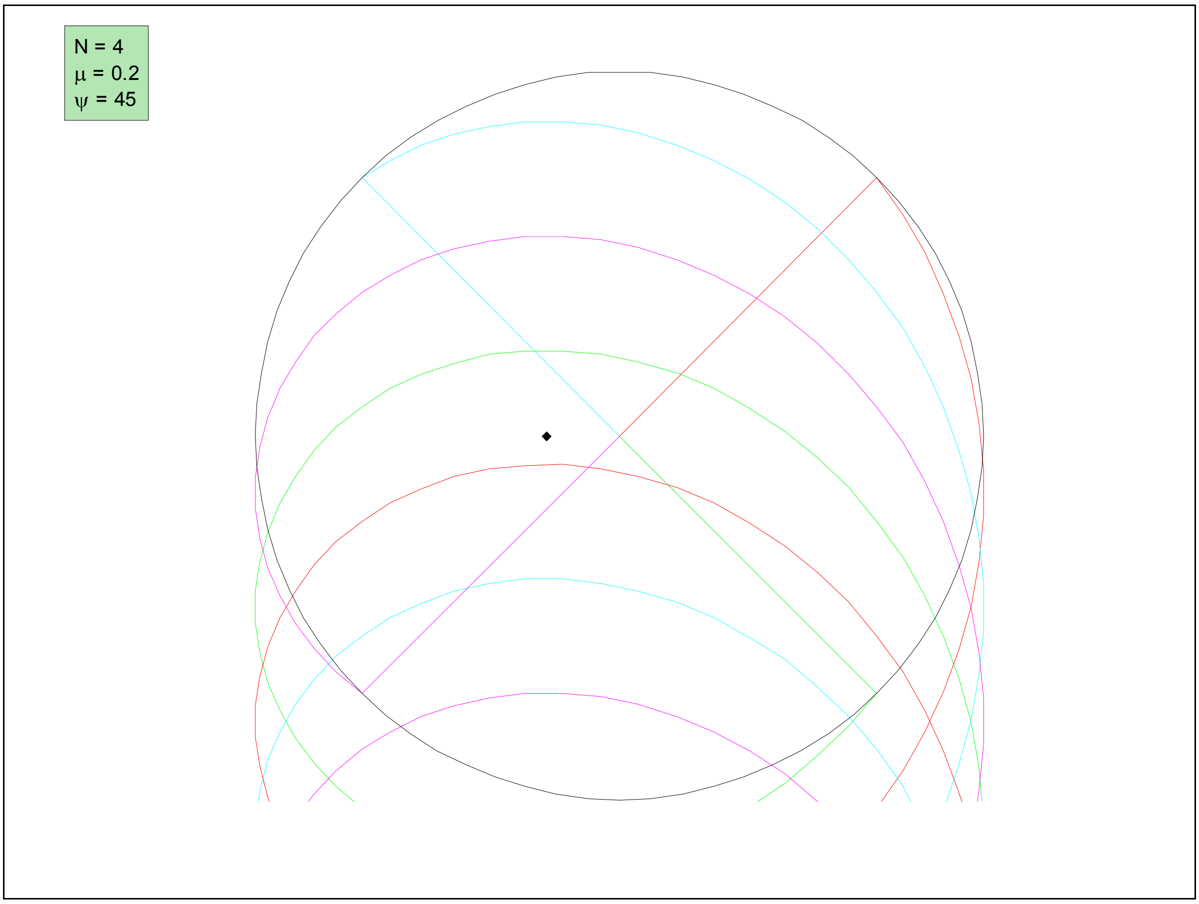

Figure 4 – An Examples of Vortex Trails: N = 4, μ = 0.2, Ψ_B = 45°

As an example, Figure 4 shows a typical set of vortex trails from a four bladed rotor at an advance ratio of 0.2. The rotor is set with the reference blade (green) at an azimuth of 45°. The colour code for the blades is:

- Green
- Red
- Cyan
- Magenta

An analysis of the geometry of the vortex trajectories now follows.

Coordinate System & Analysis

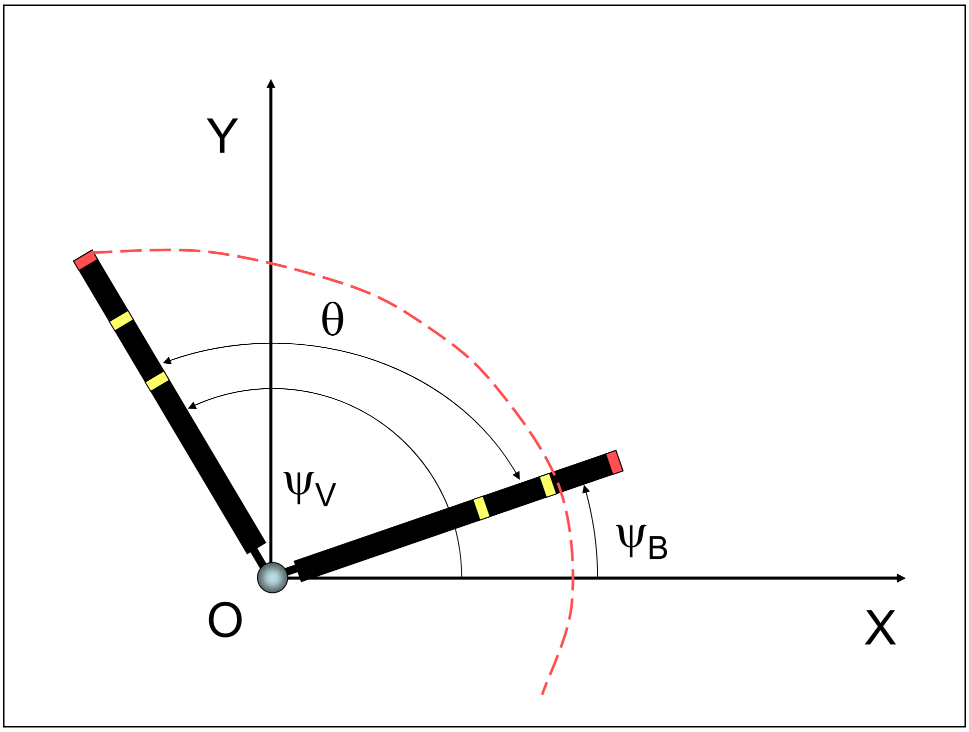

Figure 5 - Coordinate System

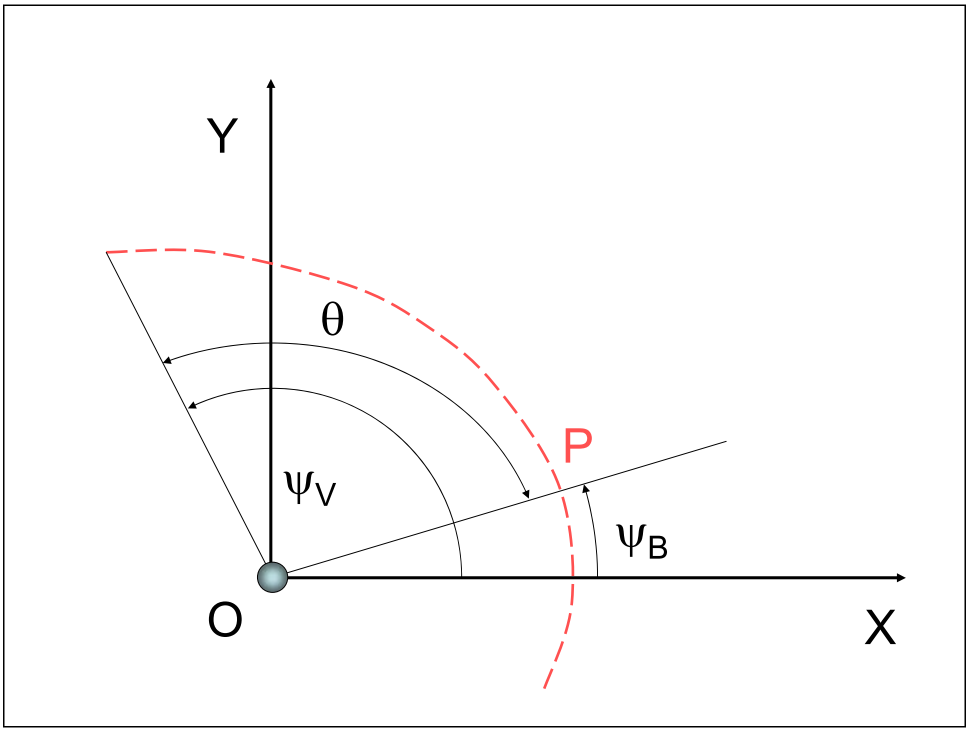

Figure 6 - Intersection of Blade & Vortex

With reference to Figure 5, the vortex is generated by the tip of the blade at azimuth angle Ψ_V and intersects the blade at azimuth angle Ψ_B at point P, as shown in Figure 6. The inter-blade angle (that between successive blades) is given by θ. The following relationship applies:

$$\Psi_V = \Psi_B + \theta \qquad (4.)$$

The radial station x corresponds to OP. *(It should be noted that the rotor is scaled to unit radius).*

From this, the point P is defined by the blade:

$$P_B = \left(x\cos\Psi_B, x\sin\Psi_B\right) \qquad (5.)$$

And by the vortex trail (where ϕ represents the rotor rotation angle):

$$\begin{aligned} P_V &= \left(\mu\phi + \cos(\Psi_V - \phi), \sin(\Psi_V - \phi)\right) \\ &= \left(\mu\phi + \cos(\Psi_B + \theta - \phi), \sin(\Psi_B + \theta - \phi)\right) \end{aligned} \qquad (6.)$$

Equating coordinates (the condition for a blade / vortex intersection) results in the following two equations:

$$x\cos\Psi_B = \mu\phi + \cos\Psi_B \cdot \cos(\theta - \phi) - \sin\Psi_B \cdot \sin(\theta - \phi) \qquad (7.)$$

and

$$x\sin\Psi_B = \sin\Psi_B \cdot \cos(\theta - \phi) + \cos\Psi_B \cdot \sin(\theta - \phi) \qquad (8.)$$

In order to calculate the conditions for an intersection we combine the equations (7) & (8) thus:

$$(7) \cdot \sin\Psi_B = (8) \cdot \cos\Psi_B \qquad (9.)$$

This on simplification reduces to:

$$\mu\phi \cdot \sin\Psi_B - \sin(\theta - \phi) = 0 \qquad (10.)$$

The value of ϕ satisfying (10) is the effective age of the vortex (expressed in rotor rotation). From this the spanwise intersection can be evaluated from equations (7) or (8) thus:

$$\begin{aligned} x &= \frac{\mu\phi + cos\left(\Psi_B + \theta - \phi\right)}{cos\,\Psi_B} \\ &= \frac{sin\left(\Psi_B + \theta - \phi\right)}{sin\,\Psi_B} \end{aligned} \quad (11.)$$

Both equations are required as conditions can occur where one of the denominators vanishes.

Examples

Examples of the types of intersection loci obtained are presented in Figures 7 to 12 for a four bladed rotor with advance ratios varying in value from 0.05 to 0.5:

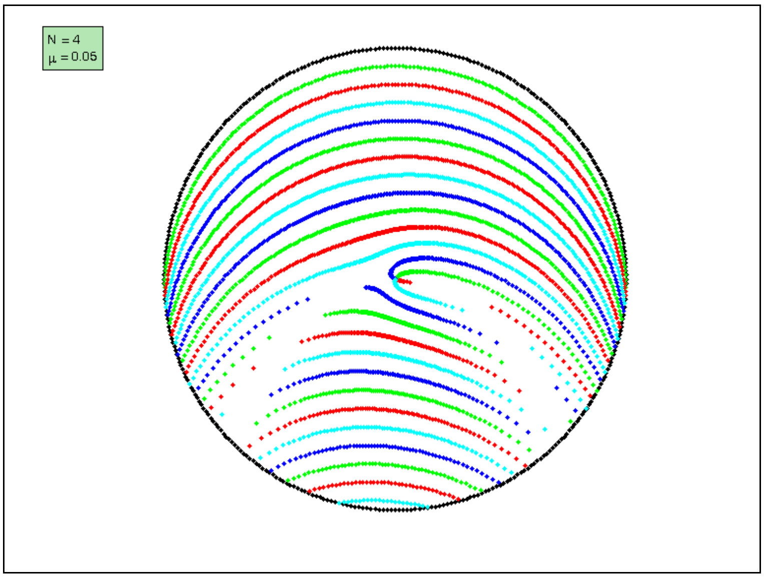

Figure 7 - N=4, μ=0.05

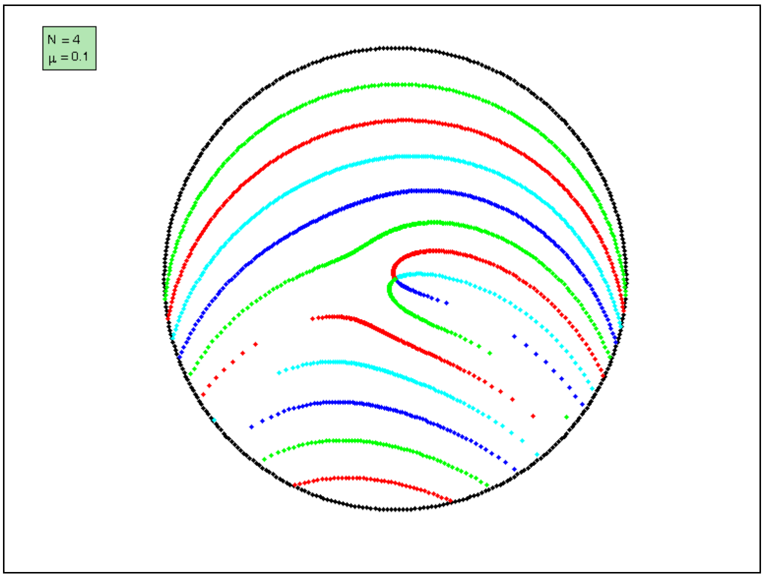

Figure 8 - N=4, μ=0.1

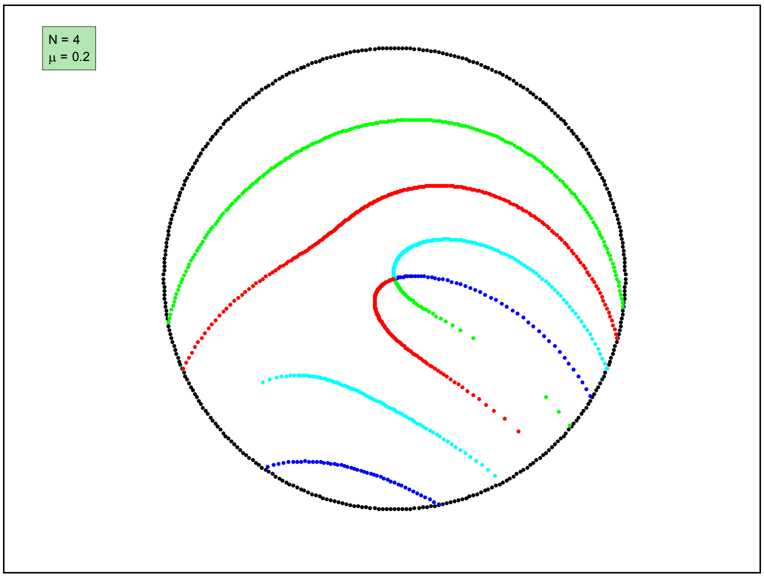

Figure 9 - N=4, μ=0.2

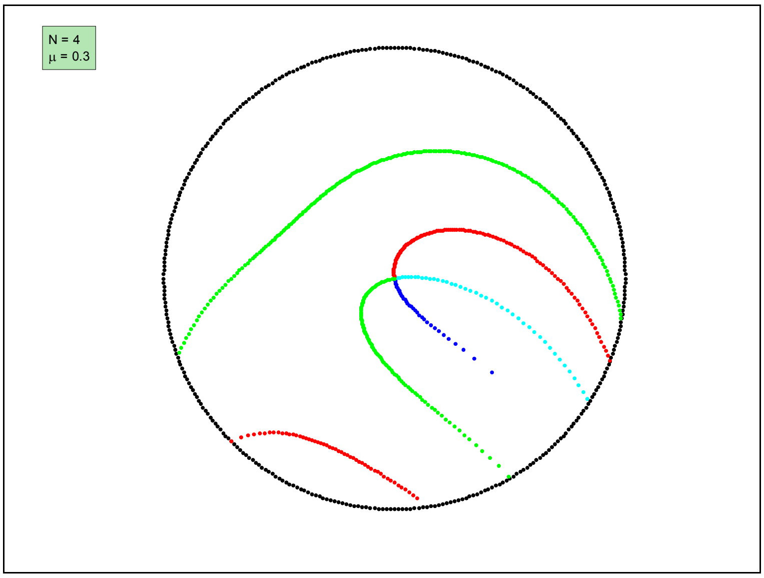

Figure 10 - N=4, μ=0.3

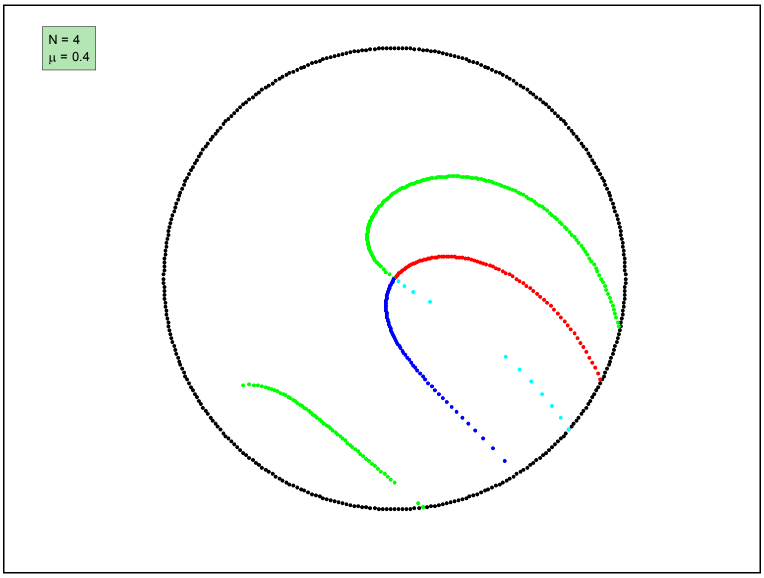

Figure 11 - N=4, μ=0.4

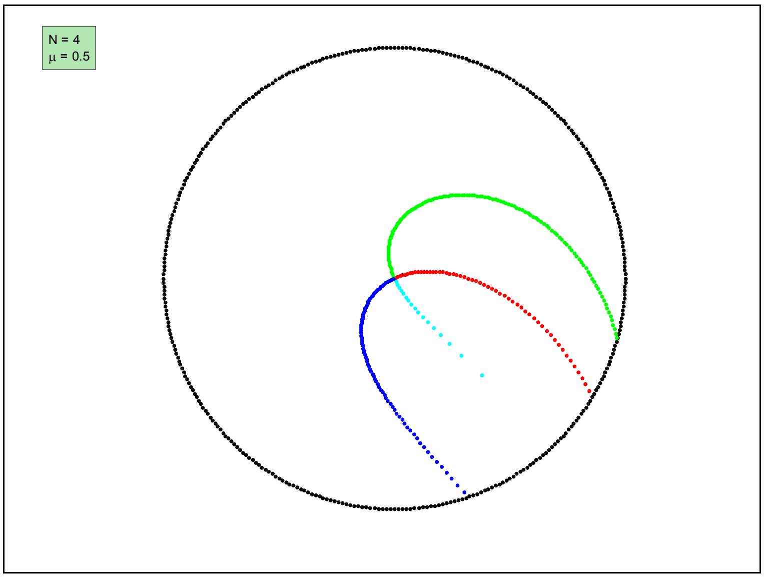

Figure 12 - N=4, μ=0.5

As can be seen, the intersections form three main types, namely:

1. At the front of the rotor disc
2. Aft on the advancing side
3. Aft of the rotor disc

These are shown in Figure 13:

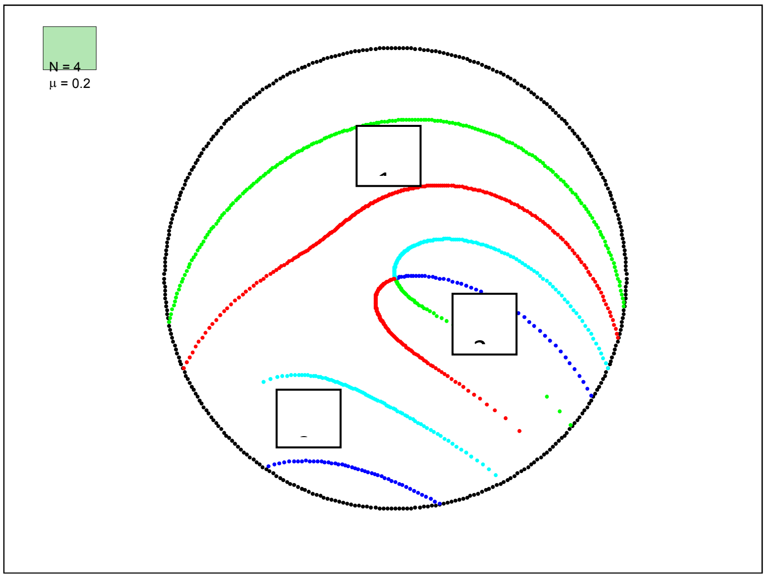

Figure 13 – Types of Blade Vortex Intersection

An interesting phenomenon is observed at particular advance ratios. This is highlighted in the following figure:

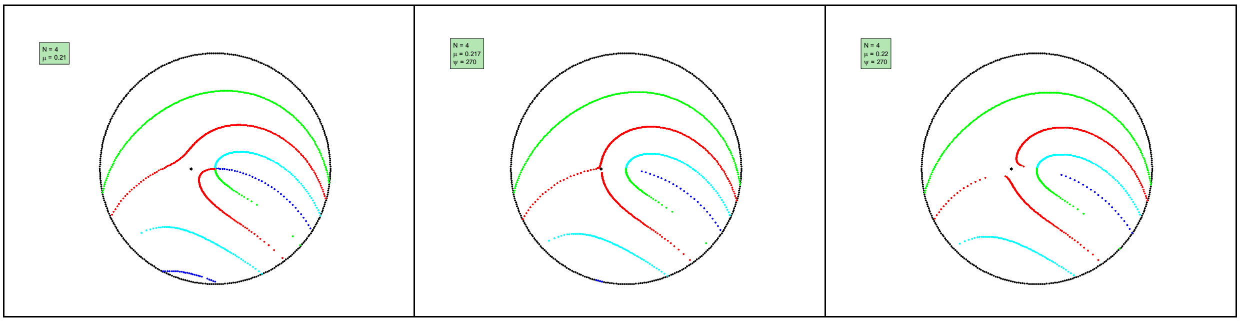

Figure 14 – Types of Blade Vortex Intersection

The rotor has four blades and the advance ratios are 0.210, 0.217 and 0.220 respectively. As can be seen, the red locus of blade vortex intersections changes from a type 1 intersection to type 2 intersection together with a type 2 changing to a type 3. The point of change is the instantaneous centre of rotation which is plotted on the figures. This indicates that the reason for this change in characteristics occurs when there is an intersection with the blade at an azimuth of 270° and the point of intersection at a radius of $x=\mu$. As previously mentioned, the vortex will lie directly laterally and therefore along the blade with a tangent condition.

This is shown in Figure 15:

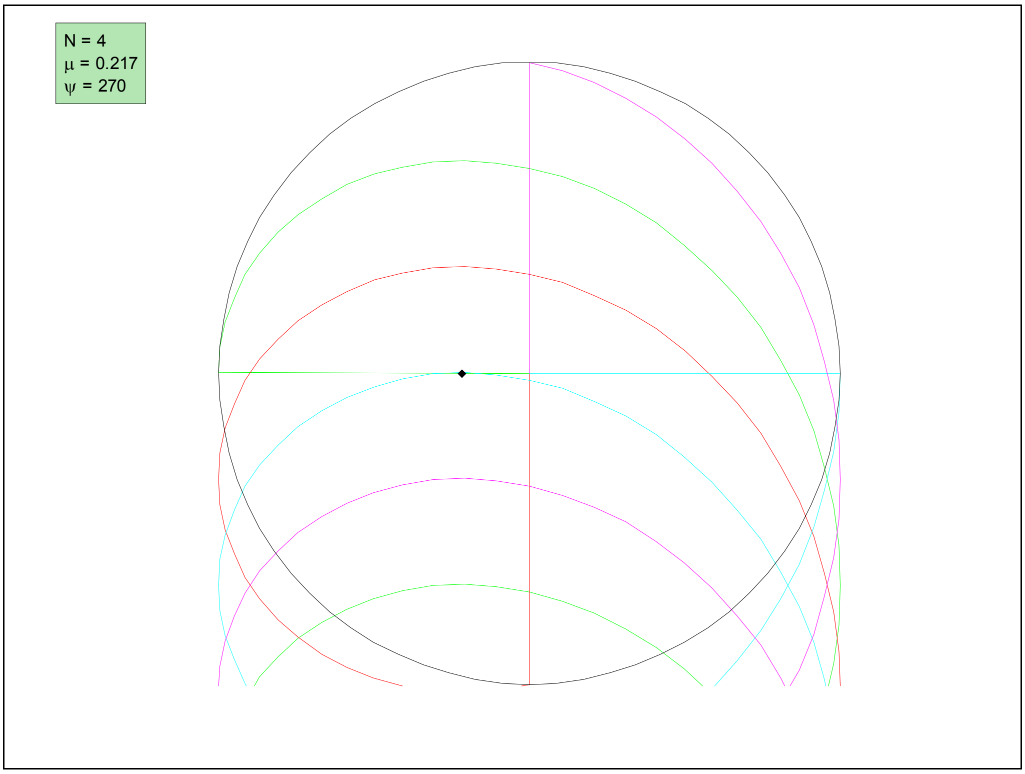

Figure 15 – Tangential Blade Vortex Intersection

The question is posed as to under what conditions will this situation occur?

The following figure (16) shows the wake geometry for this condition to be satisfied:

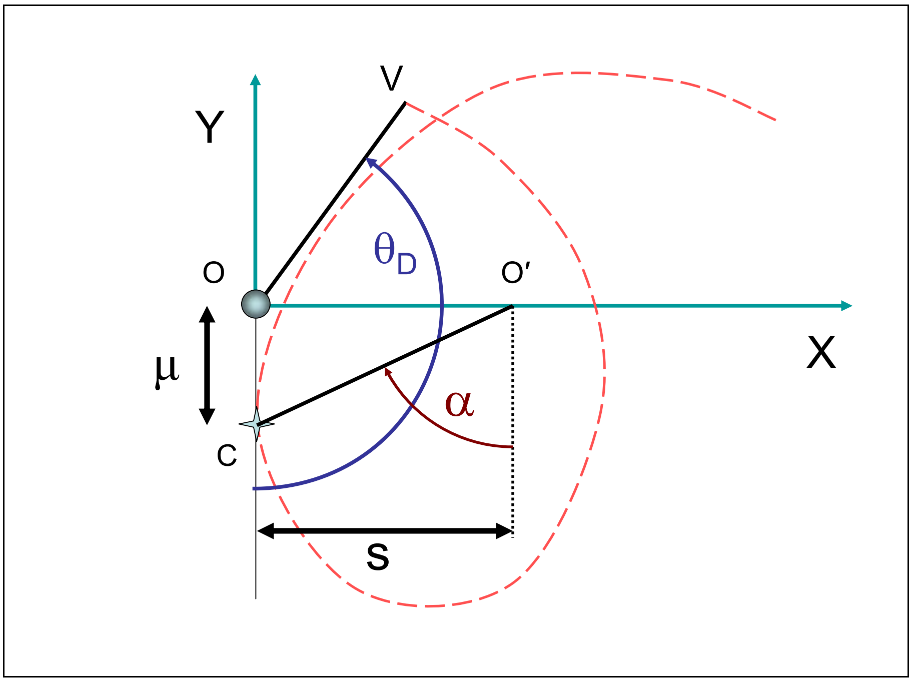

Figure 16 – Condition for Vortex Tangency

The vortex tangential intersection is shown (C) and the vortex is assumed to be generated by a blade tip (V) whose azimuthal alignment (∠ VOC) to the blade at 270° is given by θ_D. With time being reversed, the blade at OV rotates to a position O′C. It has rotated through an angle of:

$$\theta_D + \alpha \tag{12.}$$

Therefore the distance moved by the rotor centre is:

$$s = \mu(\theta_D + \alpha) \tag{13.}$$

Since OO′ is equal to unity we also have by applying Pythagoras to Triangle OO′C:

$$s^2 + \mu^2 = 1 \tag{14.}$$

Finally using the result:

$$\cos(\alpha) = \mu \tag{15.}$$

The relationship defining the blade alignment angle θ_D is given by:

$$\theta_D = \sqrt{\frac{1}{\mu^2} - 1} - \cos^{-1}(\mu) \tag{16.}$$

A plot of this equation is shown in Figure 17:

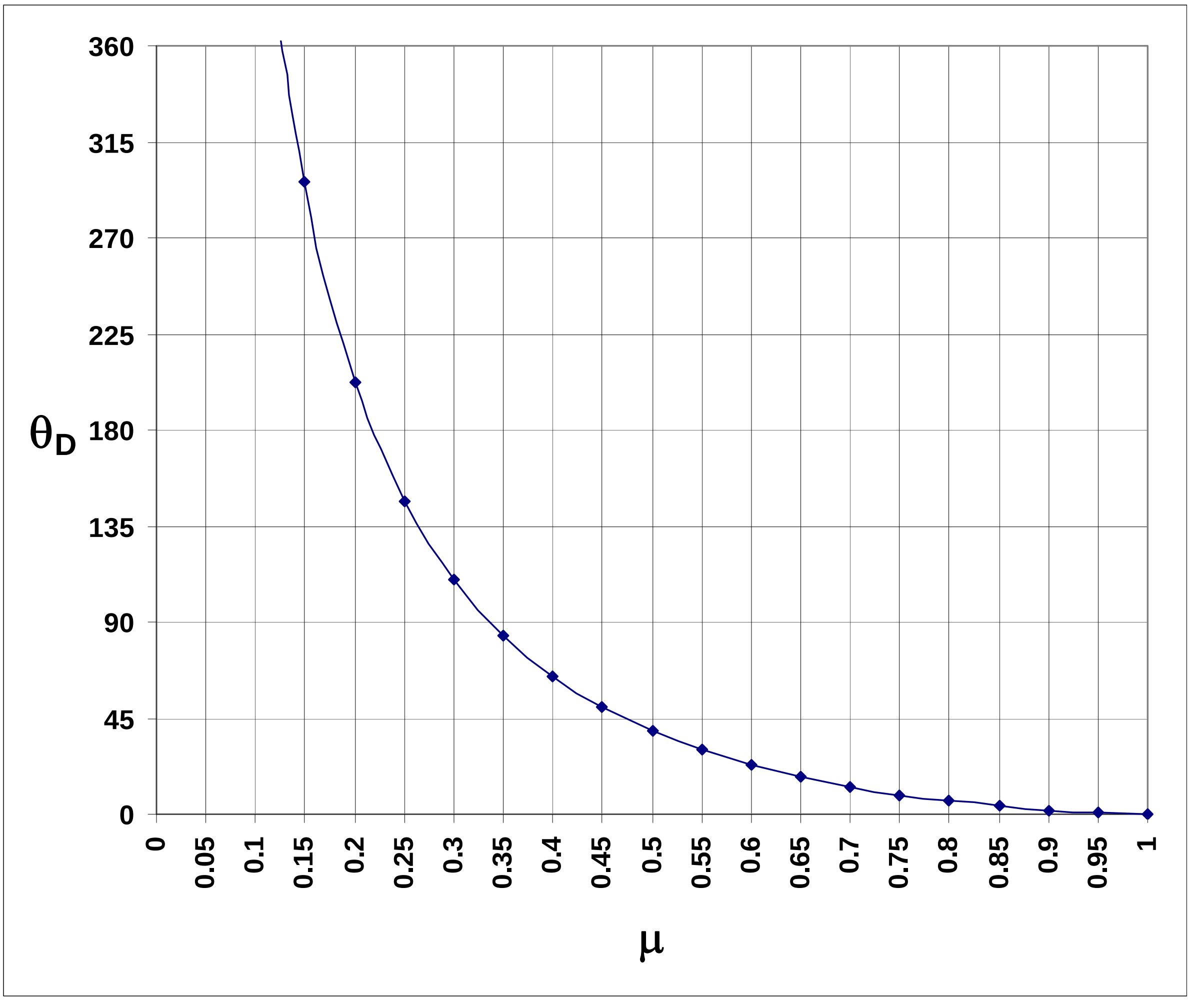

Figure 17 - Vortex Tangency Condition

Of more use are the advance ratios that give values to θ_D appropriate to a given number of blades and the subsequent azimuthal alignment between the blades.

The following table shows the advance ratios which give such conditions for tangency. N_B refers to the number of blades. N_V refers to the number of blades between that forming the vortex and that blade where the tangency interaction with the vortex occurs; *(measured in the direction of rotation)*:

μ	θ_D	N_B	N_V
0.284	120	3	1
0.176	240	3	2
0.128	360	3	3
0.337	90	4	1
0.217	180	4	2
0.161	270	4	3
0.128	360	4	4
0.379	72	5	1
0.253	144	5	2
0.191	216	5	3
0.153	288	5	4
0.128	360	5	5
0.415	60	6	1
0.284	120	6	2
0.217	180	6	3
0.176	240	6	4
0.149	300	6	5
0.128	360	6	6

Table 1 – Advance Ratio Values for BVI Tangency

For example the second row shows that at an advance ratio of 0.176, a three bladed rotor will give the condition of tangency for the vortex coming from two blades ahead. This is shown in Figure 18:

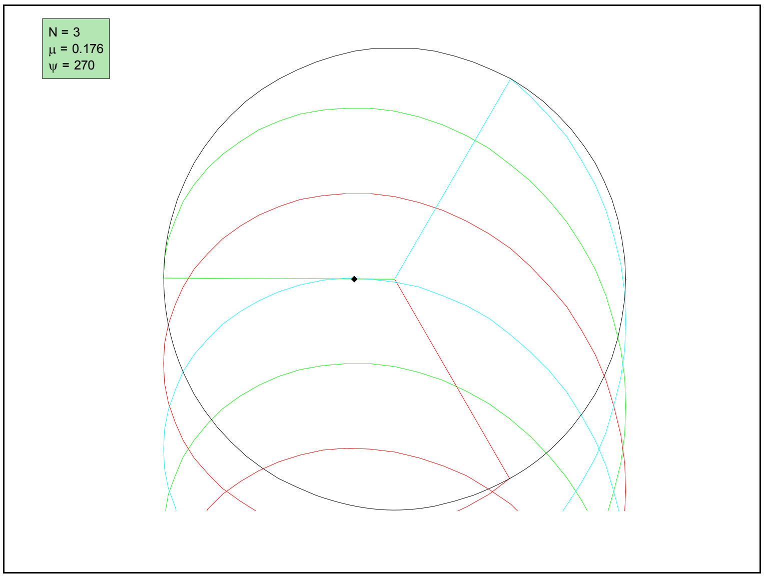

Figure 18 – Tangency Condition for N=3 & μ=0.176

It is apparent that the vortex intersections at the front of the disc will have to move towards the back of the disc as the advance ratio increases. The manner in which this is achieved is that one of the intersections of Type 1 migrate rearwards until the critical advance ratio is achieved at which point they transfer to types 2 & 3. These then smoothly move rearwards, finally clearing the rotor disc at high advance ratios.

Vortex Intersections in 3D

The above discussion gives the blade / vortex intersections in the disc plane (i.e. 2D). As a final examination of how the intersection points vary, it is of interest to consider the link between the vortex intersection position in the rotor plane and the vortex age which will give an indication of the vortex strength *(the ability of the vorticity to dissipate – with time - under viscous action)*. The vortex age is given by the angle ϕ, therefore the intersections can be plotted in 3D using the age variable (ϕ) as the third coordinate.

Examples

The following figures relate to a four bladed rotor with a range of advance ratios, including the tangency case of μ = 0.217. *(The plan view is that already discussed in the 2D interpretation, the vortex age is the vertical coordinate.)*

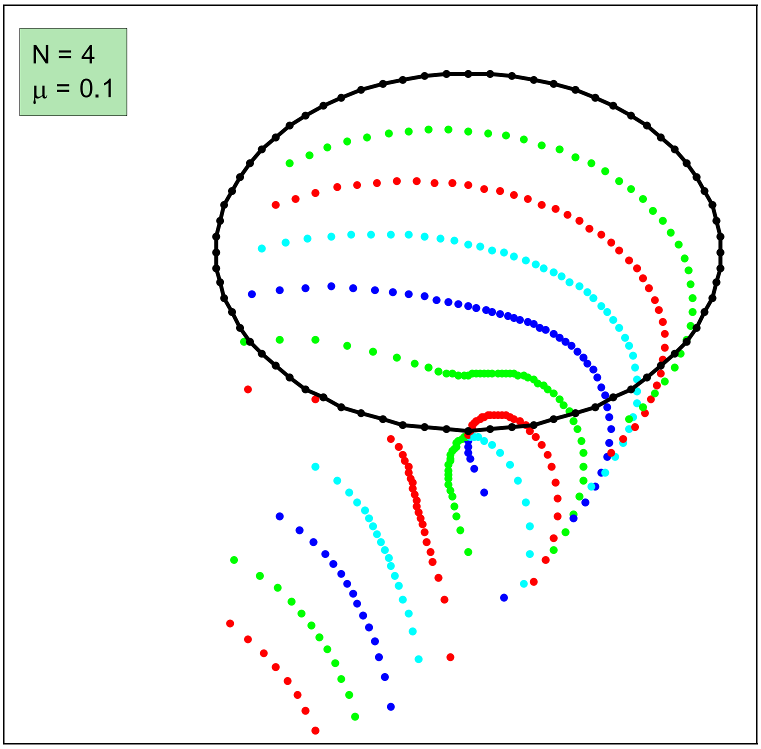

Figure 19 - N=4, μ=0.1

At the advance ratio of 0.1, there is a considerable number in intersections at the front of the rotor disc with a low age and therefore with relatively strong vorticity. As the intersection points move rearwards, in general the vortex ages increase, which would indicate a less intense interaction as the vorticity would have more time to dissipate.

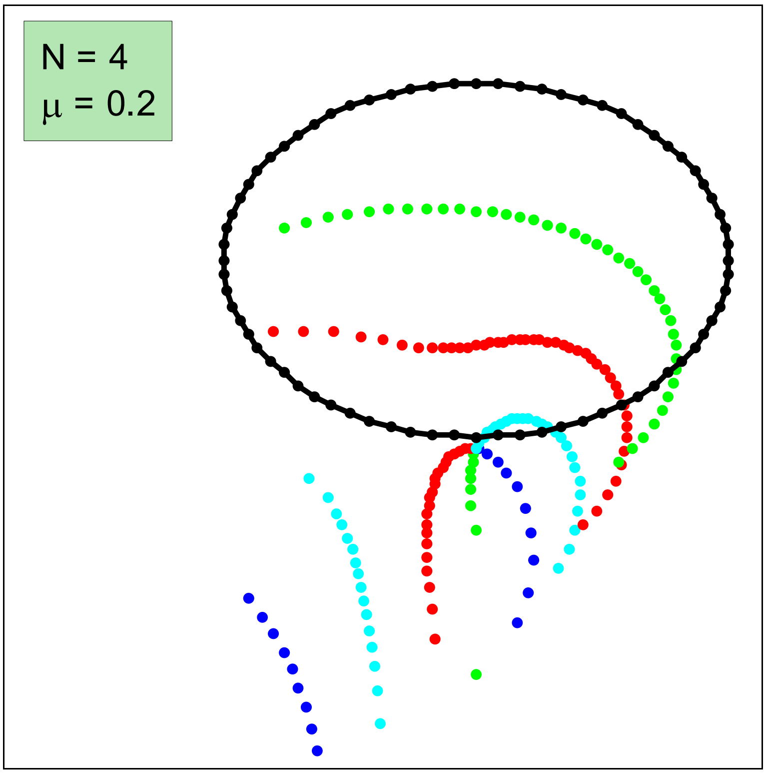

Figure 20 - N=4, μ=0.2

As the advance ratio increases to 0.2, the number of intersections at the front of the rotor disc decreases.

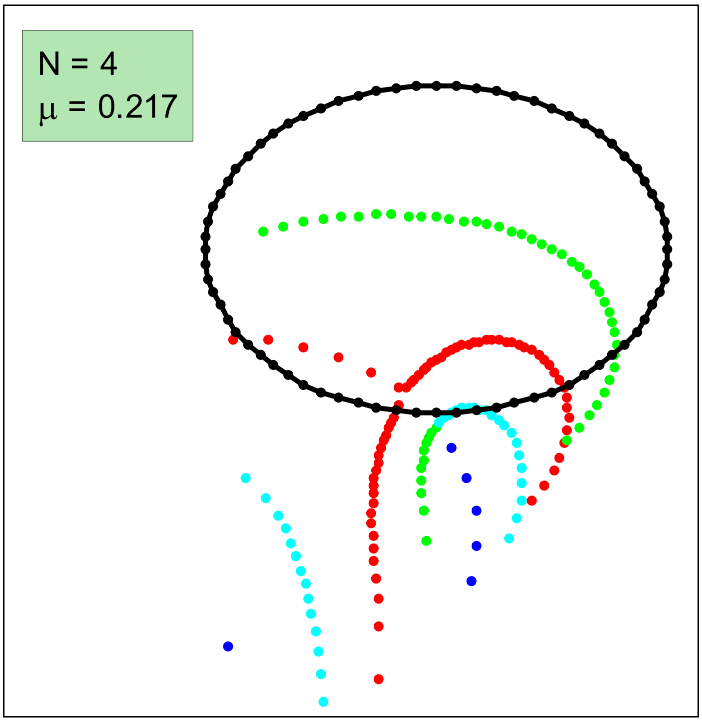

Figure 21 - N=4, μ=0.217

With an advance ratio of 0.217, the tangential blade vortex intersection occurs. This results in the coalescence of two intersection loci.

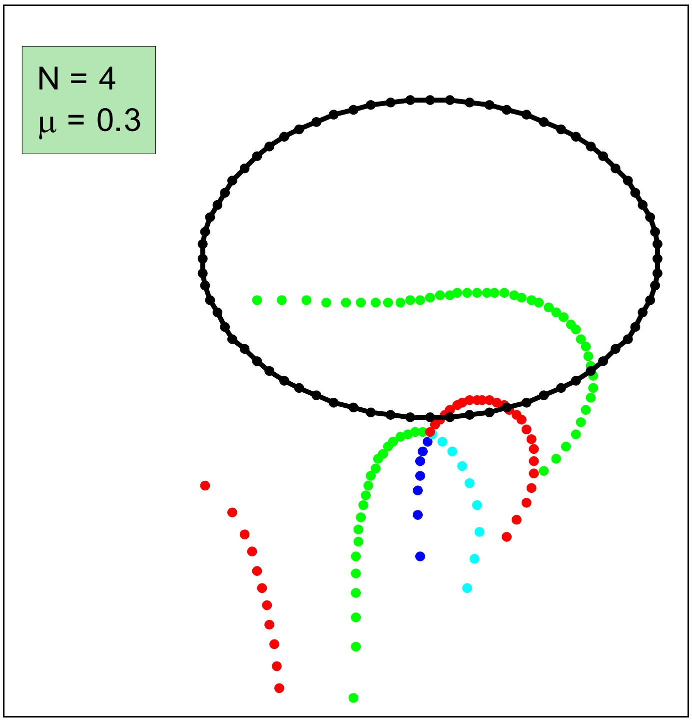

Figure 22 - N=4, μ=0.3

An advance ratio of 0.3 shows the increasing forward speed of the rotor resulting in the vortex intersections becoming scarcer.

Overall, as can be seen from the figures, the type 1 intersections form the early cases followed in turn by type 2 & type 3. As the critical tangency advance ratios are encountered the transformation between the types can be seen. The 3D presentation illustrates how the age of the vortices increases as the interaction points move towards the rear of the rotor disc.

Appendix – The Cycloid

The cycloid curve is generated by a point of a circle which is rolling (without slipping) along a straight edge. The shape of the curve is dependent on the position of the generating point relative to the centre of the circle.

Figure 23 shows the generation of a cycloid with three points P_1 (inside the circle), P_2 (on the circle) & P_3 (outside the circle). They give rise to three distinct types of cycloid, namely: curtate, conventional and prolate, as shown in Figures 24 - 26 respectively.

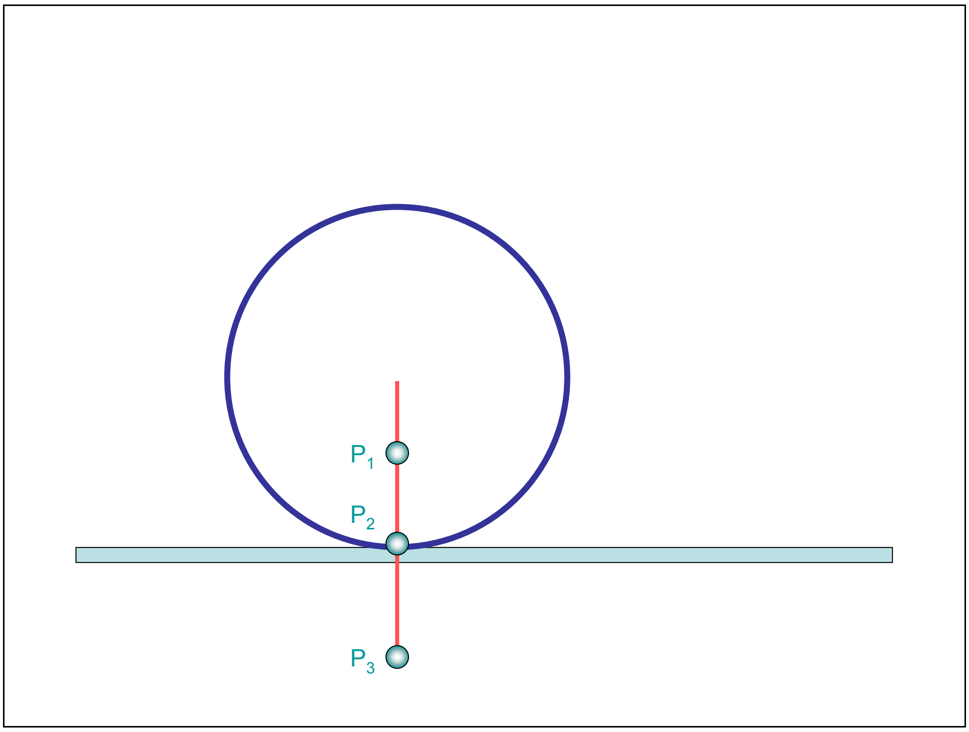

Figure 23 - Generation of Cycloids

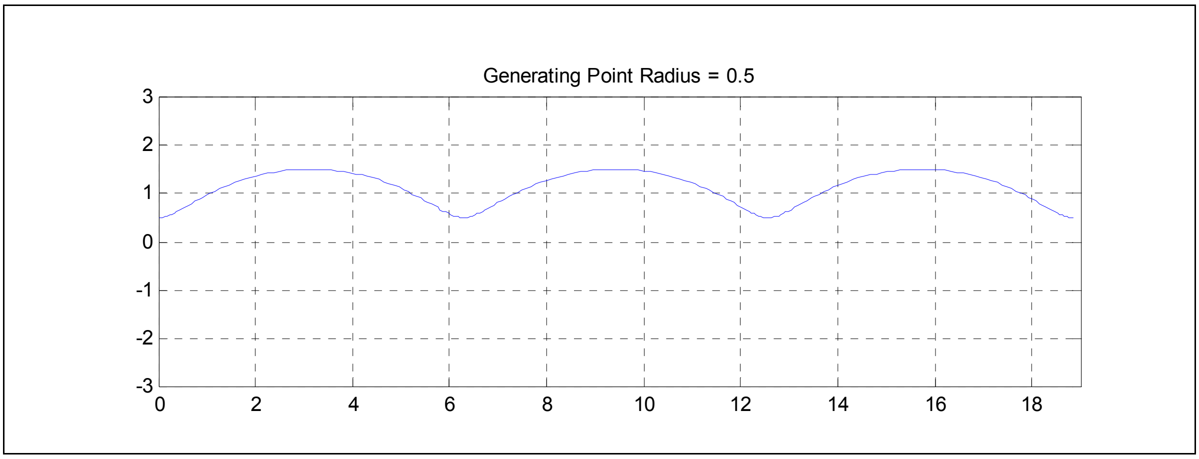

Figure 24 - Cycloid (Curtate) - P_1

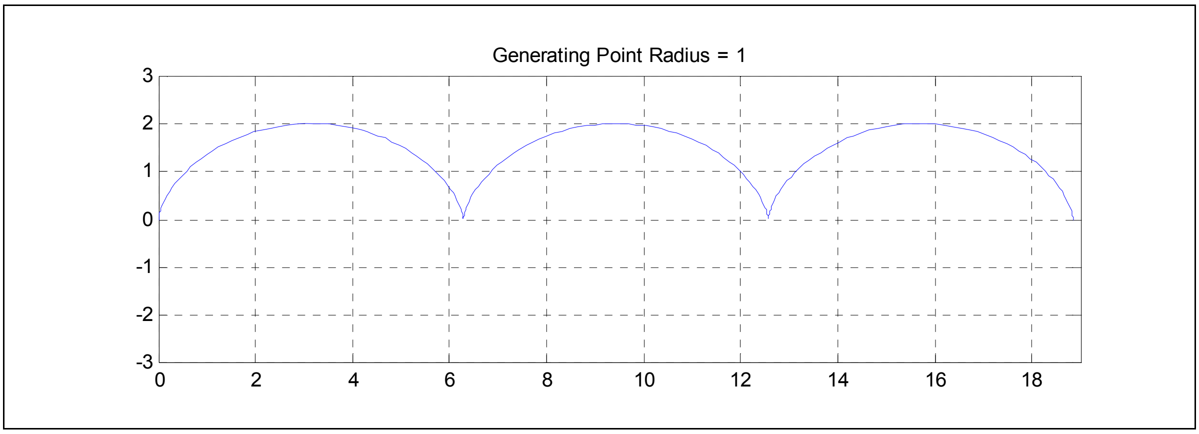

Figure 25 - Cycloid (Conventional) – P_2

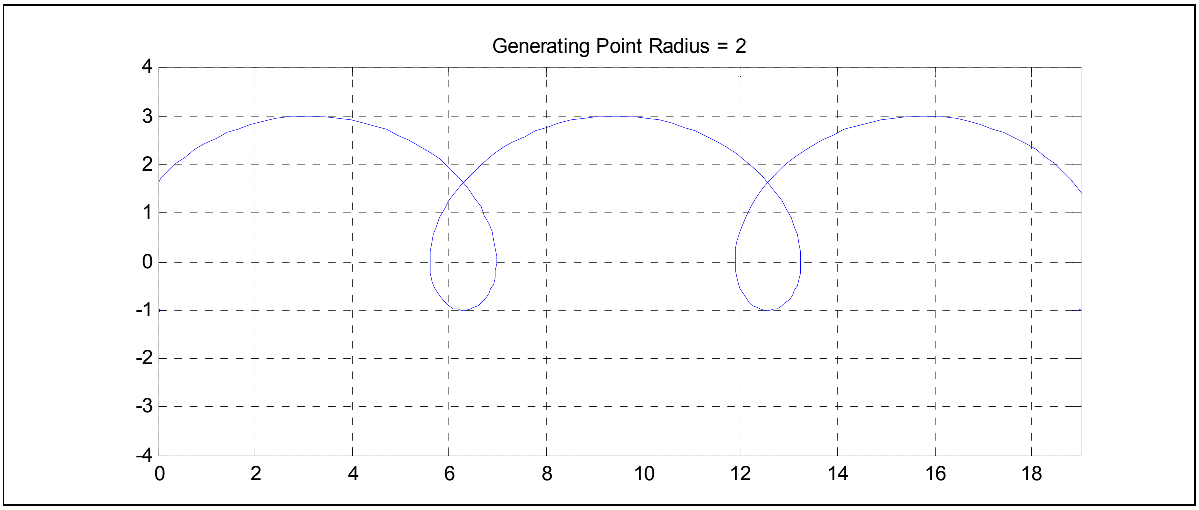

Figure 26 - Cycloid (Prolate) – P_3

In the case of a rotor in forward flight, if P_3 is at the blade tip, then the circle is of radius μ. Therefore, P_1 refers to a point of the blade with radial location equal to less than the advance ratio. P_2 refers to a blade position equal to the advance ratio and P_3 a point outside of P_2.

With advance ratios normally being less than 0.5, the blade tip will lie outside of the circle and so the vortex trails will be prolate cycloids.

MATLAB Files

Vortex Trail – 2D – Single Shot

Straight Blade

```
%
%   Programme Vortex Trail - Single Shot - Use psib for Rotor Position
%
colordef black
icol=['b','g','r','c','m','y'];
nblade=3;
psib=270;
psibr=psib*pi/180;
mu=.176;
dphi=pi/30;
phimax=2/mu;
%-----------------------------------------------------
% Calculate Points to Draw Disc Perimeter
psidisc=linspace(0,2*pi,73);
xdisc=cos(psidisc);
ydisc=sin(psidisc);
%-----------------------------------------------------
    clf
    axis([-1,1,-1,1])
      axis square
      axis manual
    axis off
      hold on
      for itheta=1:nblade
            theta=(itheta-1)*2*pi/nblade;
            x=[0];
            y=[0];
            phi=0;
            while phi<phimax
             xv=mu*phi+cos(psibr+theta-phi);
             yv=sin(psibr+theta-phi);
             x=[x,xv];
             y=[y,yv];
             phi=phi+dphi;
            end
%-----------------------------------------------------
% Plot Vortex Trail
            plot(y,-x,icol(1+mod(itheta,6)))
      end
%-----------------------------------------------------
    plot(xdisc,ydisc,'w','linewidth',1);% Draw Disc Perimeter
```

```
%-----------------------------------------------------
% Draw Centre of Local Rotation

plot(-
mu,0,'d','MarkerEdgeColor','w','MarkerFaceColor','w','MarkerSize',5);

str1(1)={['N = ',num2str(nblade)]};
str1(2)={['\mu = ',num2str(mu)]};
str1(3)={['\psi = ',num2str(psib)]};

text(-1.5,1,str1,'Color','k','BackgroundColor',[.7 .9
.7],'EdgeColor','k','Margin',5);
%-----------------------------------------------------
```

Vortex Trail – 2D – Movie – Rotor View

Straight Blade

```
%
%   Programme Vortex Trail
%
colordef black
icol=['b','g','r','c','m','y'];
nblade=4;
dpsib=1;
dpsibr=dpsib*pi/180;
nframe=360;
mu=.337;
dphi=pi/30;
phimax=2/mu;
%-----------------------------------------------------
% Calculate Points to Draw Disc Perimeter
psidisc=linspace(0,2*pi,73);
xdisc=cos(psidisc);
ydisc=sin(psidisc);
%-----------------------------------------------------
for iframe=1:nframe
    clf
    axis([-1,1,-1,1])
      axis square
      axis manual
    axis off
      hold on
    psibr=(iframe-1)*dpsibr;
      for itheta=1:nblade
            theta=(itheta-1)*2*pi/nblade;
            x=[0];
            y=[0];
            phi=0;
            while phi<phimax
             xv=mu*phi+cos(psibr+theta-phi);
             yv=sin(psibr+theta-phi);
             x=[x,xv];
             y=[y,yv];
             phi=phi+dphi;
            end
%-----------------------------------------------------
% Plot Vortex Trail
            plot(y,-x,icol(1+mod(itheta,6)))
      end
%-----------------------------------------------------
    plot(xdisc,ydisc,'w','linewidth',1);% Draw Disc Perimeter
    text(1,1,['Mu = ',num2str(mu)]);
```

```
%----------------------------------------------------
% Draw Centre of Local Rotation

    plot(-
mu,0,'d','MarkerEdgeColor','w','MarkerFaceColor','w','MarkerSize',5);
%----------------------------------------------------
    M(iframe)=getframe(gcf);
end
%----------------------------------------------------
%movie2avi(M,'fred','compression','none');
movie2avi(M,'fred','compression','Cinepak');
%movie(M)
```

Vortex Trail – 2D – Movie – Rotor View

Cranked Blade

```
%
%   Programme Vortex Trail - Cranked Blade Tip
%
colordef black
icol=['b','g','r','c','m','y'];
nblade=4;
dpsib=1;
dpsibr=dpsib*pi/180;
nframe=360;
mu=.1;
dphi=pi/30;
phimax=2/mu;
%-----------------------------------------------------
% Set Up Blade Tip Crank Parameters
xcrnk=.8;
alfcrnkd=30;
alfcrnkr=alfcrnkd*pi/180;
swp1=(1-xcrnk)*tan(alfcrnkr);
%-----------------------------------------------------
% Calculate Points to Draw Disc Perimeter
psidisc=linspace(0,2*pi,73);
rdisc=sqrt(1+swp1^2);
xdisc=rdisc*cos(psidisc);
ydisc=rdisc*sin(psidisc);
%-----------------------------------------------------
for iframe=1:nframe
    clf
    axis([-rdisc,rdisc,-rdisc,rdisc])
      axis square
      axis manual
    axis off
      hold on
    psibr=(iframe-1)*dpsibr;
      for itheta=1:nblade
            theta=(itheta-1)*2*pi/nblade;
            x=[0,xcrnk*cos(psibr+theta)];
            y=[0,xcrnk*sin(psibr+theta)];
            phi=0;
            while phi<phimax
             xv=mu*phi+cos(psibr+theta-phi)+swp1*sin(psibr+theta-phi);
             yv=sin(psibr+theta-phi)-swp1*cos(psibr+theta-phi);
             x=[x,xv];
             y=[y,yv];
             phi=phi+dphi;
            end
```

```
%---------------------------------------------------
% Plot Vortex Trail
           plot(y,-x,icol(1+mod(itheta,6)))
     end
%---------------------------------------------------
% Draw Disc Perimeter
    plot(xdisc,ydisc,'w','linewidth',1);
%---------------------------------------------------
% Draw Text Box
    str1=['\mu = ',num2str(mu)];
    s=char(str1);
      text(1,1,s,'Fontsize',16,'Color','black','FontWeight','Bold','Bac
kgroundColor',[.7 .9 .7],'Edgecolor','black');
%---------------------------------------------------
% Draw Centre of Local Rotation
    plot(-
mu*rdisc,0,'d','MarkerEdgeColor','w','MarkerFaceColor','w','MarkerSize'
,5);
%---------------------------------------------------
    M(iframe)=getframe(gcf);
end
%---------------------------------------------------
%movie2avi(M,'fred','compression','none');
%movie2avi(M,'fred','compression','Cinepak');
%movie(M)
```

BVI – 2D – Single Shot

```
%
%   Program BVI 2D - Single Shot
%
%   SJN 27/1/08
%
icol=['b','g','r','c','m','y'];
nblade=3;
npsib=360;
mu=0.176;
phimax=2/mu;
dphitol=0.00001;
hold on
axis([-1,1,-1,1])
for ipsib=1:npsib % Control Point Blade Azimuth Stepping
psib=(ipsib-1)*360/npsib;
psibr=psib*pi/180;
    for itheta=1:nblade % Vortex Blade Azimuth Stepping
      theta=(itheta-1)*2*pi/nblade;
      phi=0;
      dphi=pi/9;
            while phi<phimax % Moving Down Vortex Trail
            if phi==0 & theta==0 % Trivial Root
                x=1;
                S=['k','.'];
                xplt=x*cos(psibr);
                yplt=x*sin(psibr);
                plot(yplt,-xplt,S);
            end
            fold=mu*phi*sin(psibr)-sin(theta-phi);
            while abs(dphi)>dphitol %Iteration within Tolerance?
                phi=phi+dphi;
% ???????????????????????????????????????????
                if phi>phimax
                    break % Break Out of Inner While Loop when Far
Enough Down Vortex Trail
                end
% ???????????????????????????????????????????
                fnew=mu*phi*sin(psibr)-sin(theta-phi);
                if fold*fnew<0 % Stepped over Root?
                    dphi=-0.1*dphi;
                end
                fold=fnew;
            end %Iteration within Tolerance?

% ???????????????????????????????????????????
            if phi>phimax
                    break % Break Out of Outer While Loop when Far
Enough Down Vortex Trail
                end
```

```
% ???????????????????????????????????????????
            if abs(cos(psibr))>0.5

                x=(mu*phi+cos(psibr+theta-phi))/cos(psibr);
            else
                x=sin(psibr+theta-phi)/sin(psibr);
            end
            dphi=pi/9;
            phi=phi+dphi;
            if x*(x-1)<0
                xplt=x*cos(psibr);
                yplt=x*sin(psibr);
                S=[icol(mod(itheta,6)),'.'];
                plot(yplt,-xplt,S')
            end
           end % Moving Down Vortex Trail
      end % Vortex Blade Azimuth Stepping
end % Control Point Blade Azimuth Stepping
axis equal
axis off

xcor=-mu;
ycor=0;
plot(xcor,ycor,'d','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerS
ize',5);% Draw Centre of Local Rotation

str1(1) = {['N = ',num2str(nblade)]};
str1(2) = {['\mu = ',num2str(mu)]};

text(-1.5,1,str1,'BackgroundColor',[.7 .9
.7],'EdgeColor','k','Margin',5);
hold off
```

BVI – 2D – Movie

```
%
%   Program BVI2MOV
%
%   Plan View of Rotor Disc & BVI - for a range of Advance Ratios - +
Movie
%   Creation
%
%   SJN 27/1/08
%
colordef black;
icol=['b','g','r','c','m','y'];
nblade=4;
npsib=72;
nframe=50;
dphitol=0.00001;
axis([-1,1,-1,1])
for iframe=1:nframe % Movie Frame Loop
    clf
        psidisc=linspace(0,2*pi,73); % Draw Disc Perimeter
        xdisc=cos(psidisc);
        ydisc=sin(psidisc);
        plot(xdisc,ydisc,'w','linewidth',2);% Draw Disc Perimeter
        hold on
    mu=iframe/nframe*.5;
        xcor=-mu;
        ycor=0;

plot(xcor,ycor,'d','MarkerEdgeColor','w','MarkerFaceColor','w','MarkerS
ize',5);% Draw Centre of Local Rotation
          hold on
    phimax=2/mu;
      for ipsib=1:npsib % Control Point Blade Azimuth Stepping
        psib=(ipsib-1)*360/npsib;
          psibr=psib*pi/180;
        for itheta=1:nblade % Vortex Blade Azimuth Stepping
           theta=(itheta-1)*2*pi/nblade;
           phi=0;
           dphi=pi/9;
                 while phi<phimax % Moving Down Vortex Trail
                if phi==0 & theta==0 % Trivial Root
                    x=1;
                    S=['w','.'];
                    xplt=x*cos(psibr);
                    yplt=x*sin(psibr);
                    hold on
                    plot(yplt,-xplt,S);
                end
                fold=mu*phi*sin(psibr)-sin(theta-phi);
                while abs(dphi)>dphitol %Iteration within Tolerance?
```

```
                  phi=phi+dphi;
      % ??????????????????????????????????????????
                  if phi>phimax
                      break % Break Out of Inner While Loop when Far
Enough Down Vortex Trail
                  end
      % ??????????????????????????????????????????
                  fnew=mu*phi*sin(psibr)-sin(theta-phi);
                  if fold*fnew<0 % Stepped over Root?
                      dphi=-0.1*dphi;
                  end
                  fold=fnew;
              end %Iteration within Tolerance?

      % ??????????????????????????????????????????
              if phi>phimax
                      break % Break Out of Outer While Loop when Far
Enough Down Vortex Trail
                  end
      % ??????????????????????????????????????????
              if abs(cos(psibr))>0.5

                  x=(mu*phi+cos(psibr+theta-phi))/cos(psibr);
              else
                  x=sin(psibr+theta-phi)/sin(psibr);
              end
              dphi=pi/9;
              phi=phi+dphi;
              if x*(x-1)<0
                  xplt=x*cos(psibr);
                  yplt=x*sin(psibr);
                  S=[icol(mod(itheta,6)),'.'];
                  hold on
                  plot(yplt,-xplt,S')
              end
               end % Moving Down Vortex Trail
          end % Vortex Blade Azimuth Stepping
      end % Control Point Blade Azimuth Stepping
      title(['Mu = ',num2str(mu,'%5.3g')]);
      axis equal
     axis off
      M(iframe)=getframe(gcf);
end % Movie Frame Loop
```

BVI – 3D – Single Shot

```
%
%   Program BVI 3D View - Single Shot
%
%   SJN 27/1/08
%
colordef black;
icol=['b','g','r','c','m','y'];
nblade=4;
npsib=72;
mu=0.3;
phimax=2/mu;
dphitol=0.00001;
hold on
axis([-1,1,-1,1])
psidisc=linspace(0,2*pi,73); % Draw Disc Perimeter
xdisc=cos(psidisc);
ydisc=sin(psidisc);
zdisc=zeros(1,73);
plot3(xdisc,ydisc,zdisc,'w','linewidth',2);% Draw Disc Perimeter
hold on
phicor=(1/mu^2-1)^.5;
xcor=-mu;
ycor=0;
zcor=-mu*phicor;
%plot3(xcor,ycor,zcor,'d','MarkerEdgeColor','w','MarkerFaceColor','w','Mark
erSize',5);% Draw Centre of Local Rotation
for ipsib=1:npsib % Control Point Blade Azimuth Stepping
psib=(ipsib-1)*360/npsib;
psibr=psib*pi/180;
    for itheta=1:nblade % Vortex Blade Azimuth Stepping
    theta=(itheta-1)*2*pi/nblade;
    phi=0;
    dphi=pi/9;
        while phi<phimax % Moving Down Vortex Trail
            if phi==0 & theta==0 % Trivial Root
                x=1;
                S=['k','.'];
                xplt=x*cos(psibr);
                yplt=x*sin(psibr);
                zplt=0;
                plot3(yplt,-xplt,zplt,S);
            end
            fold=mu*phi*sin(psibr)-sin(theta-phi);
            while abs(dphi)>dphitol %Iteration within Tolerance?
                phi=phi+dphi;
% ???????????????????????????????????????????
                if phi>phimax
                    break % Break Out of Inner While Loop when Far Enough
Down Vortex Trail
                end
% ???????????????????????????????????????????
```

```
                fnew=mu*phi*sin(psibr)-sin(theta-phi);
                if fold*fnew<0 % Stepped over Root?
                    dphi=-0.1*dphi;
                end
                fold=fnew;
            end %Iteration within Tolerance?

% ????????????????????????????????????????????
            if phi>phimax
                    break % Break Out of Outer While Loop when Far Enough
Down Vortex Trail
                end
% ????????????????????????????????????????????
            if abs(cos(psibr))>0.5

                x=(mu*phi+cos(psibr+theta-phi))/cos(psibr);
            else
                x=sin(psibr+theta-phi)/sin(psibr);
            end
            dphi=pi/9;
            phi=phi+dphi;
            if x*(x-1)<0
                xplt=x*cos(psibr);
                yplt=x*sin(psibr);
                zplt=-mu*phi;
                S=[icol(mod(itheta,6)),'.'];
                plot3(yplt,-xplt,zplt,S')
            end
        end % Moving Down Vortex Trail
    end % Vortex Blade Azimuth Stepping
end % Control Point Blade Azimuth Stepping
axis equal
view(50,45);
axis off
str1(1)={['N = ',num2str(nblade)]};
str1(2)={['\mu = ',num2str(mu)]};

text(-1.5,-1,0.5,str1,'Color','k','BackgroundColor',[.7 .9
.7],'EdgeColor','k','Margin',5);
hold off
```

BVI – 3D – Movie

```
%
%   Program BVI3DMOV - 3D BVI Calculation - 3rd Dimension is Vortex Age
%
%   Movie Creqation with Increasing Advance Ratio - With Vertical Line
through Instantaneous Centre of Rotation
%
%   SJN 27/1/08
%
colordef black;
icol=['b','g','r','c','m','y'];
nblade=4;
npsib=72;
nframe=5;
dphitol=0.00001;
axis([-1,1,-1,1,-1,0])
for iframe=1:nframe % Movie Frame Loop
    clf
    axis([-1,1,-1,1,-1,1])
    axis equal
    axis manual
    axis off
    hold on
    mu=iframe/nframe*.5;
    phicor=(1/mu^2-1)^.5;
        xcor(1)=-mu;
        ycor(1)=0;
        zcor(1)=0;
        xcor(2)=-mu;
        ycor(2)=0;
        zcor(2)=-2;
        plot3(xcor,ycor,zcor,'Color','y','linewidth',2);% Draw Locus of
Centre of Local Rotation
        hold on
    phimax=2/mu;
    for ipsib=1:npsib % Control Point Blade Azimuth Stepping
        psib=(ipsib-1)*360/npsib;
        psibr=psib*pi/180;
        for itheta=1:nblade % Vortex Blade Azimuth Stepping
        theta=(itheta-1)*2*pi/nblade;
        phi=0;
        dphi=pi/9;
            while phi<phimax % Moving Down Vortex Trail
                if phi==0 & theta==0 % Trivial Root
                    x=1;
                    S=['w','.'];
                    xplt=x*cos(psibr);
                    yplt=x*sin(psibr);
                    zplt=0;
                    hold on
                    plot3(yplt,-xplt,zplt,S);
                end
```

```
                fold=mu*phi*sin(psibr)-sin(theta-phi);
                while abs(dphi)>dphitol %Iteration within Tolerance?
                    phi=phi+dphi;
    % ????????????????????????????????????????????
                    if phi>phimax
                        break % Break Out of Inner While Loop when Far
Enough Down Vortex Trail
                    end
    % ????????????????????????????????????????????
                    fnew=mu*phi*sin(psibr)-sin(theta-phi);
                    if fold*fnew<0 % Stepped over Root?
                        dphi=-0.1*dphi;
                    end
                    fold=fnew;
                end %Iteration within Tolerance?

    % ????????????????????????????????????????????
                if phi>phimax
                        break % Break Out of Outer While Loop when Far
Enough Down Vortex Trail
                    end
    % ????????????????????????????????????????????
                if abs(cos(psibr))>0.5

                    x=(mu*phi+cos(psibr+theta-phi))/cos(psibr);
                else
                    x=sin(psibr+theta-phi)/sin(psibr);
                end
                dphi=pi/9;
                phi=phi+dphi;
                if x*(x-1)<0
                    xplt=x*cos(psibr);
                    yplt=x*sin(psibr);
                    zplt=-mu*phi;
                    S=[icol(mod(itheta,6)),'.'];
                    hold on
                    plot3(yplt,-xplt,zplt,S')
                end
            end % Moving Down Vortex Trail
        end % Vortex Blade Azimuth Stepping
    end % Control Point Blade Azimuth Stepping
    psidisc=linspace(0,2*pi,73); % Draw Disc Perimeter
    xdisc=cos(psidisc);
    ydisc=sin(psidisc);
    zdisc=zeros(1,73);
    plot3(xdisc,ydisc,zdisc,'w','linewidth',2);% Draw Disc Perimeter
    hold on
    str1(1)={['N = ',num2str(nblade)]};
    str1(2)={['\mu = ',num2str(mu)]};
    %text(-1.5,-1,0.5,str1,'Color','k','BackgroundColor',[.7 .9
.7],'EdgeColor','k','Margin',10);
    %axis equal
    title(str1(2));
    view(50,45); % 3D View
    %view(0,90); % Plan View
    %view(90,0); % Side View
    M(iframe)=getframe(gcf);
end % Movie Frame Loop

%movie2avi(M,'fred','compression','none');
```

Cycloid Generation

```
%
%   Cycloid Generation
%
%   SJN 27/1/08
%
n=3; %   No. of Turns
boa=1.5; %   Ratio of Point Radius to Circle Radius (<1 Curtate / =1 Normal
/ >1 Prolate)
xscale=fix(2*n*pi+1);
yscale=fix(2+1+boa);
theta=linspace(0,2*n*pi,72*n);
x=theta-boa*sin(theta);
y=1-boa*cos(theta);
plot(x,y);
axis equal;
axis([0,xscale,-yscale,yscale]);
grid on;
title(['Generating Point Radius = ',num2str(boa)]);
```