
We now look at the maths involved in describing axisymmetric flow for an incompressible, irrotational (inviscid) fluid. Such a fluid is like water, in which case the topic is called Hydrodynamics. But we will pretend that air also is incompressible, in which case the topic is called Aerodynamics. The intention is to provide the equations used for the previous article, Part 5, which describe the flow of air around the nose of an axisymmetric airplane.

The type of maths concerned is called calculus, a subject which is understood by about .0001% of the World’s population. I will make no effort to explain the calculus, unless I have a neurological fit and realise that I might understand some of it myself!

We first need an equation, which describes a uniform flow of air parallel to the abcissa (x-axis) in our rectilinear coordinate system. The stream function \( \Psi_1 \) represents the volume flow rate of such a uniform stream. It is a number, for example, we can write:

\[
\Psi_1 = 50 \text{ litres per second for air flowing past a given area of space.}
\]

This is not very general. We really want to include some extra features, such as the velocity of the flow and the area of space of interest to our problem. Here it is:

\[
\Psi_1 = -Ur^2 \sin^2 \theta / 2 = -Uy^2 / 2
\]

Here, \( U \) is the velocity of the uniform stream (think ‘wind’ or ‘airspeed’) along the abcissa. It has no other components, and represents a number, such as 50 metres per second. The other symbols are given in the diagram below, and “sin” is the trigonometric function, not something norty.
Now we need a “source” of fluid. In the previous work, the source was a point that injected fluid, into space, in a radial fashion, like the spokes on a bicycle wheel. This time, we will go for something more elegant. A point source injecting into an axisymmetric space will yield too blunt a representation of the aircraft spinner, although it could be good for the radial cowl of, say, a Sea Fury. We can get the both of both worlds by using the stream function for an axisymmetric line source. By making the line source short, we can get a blunt shape, and by making it long we can get a more pointy shape! If this is confusing, remember that we are going to combine the stream functions for a uniform stream and various sources, and even negative sources, which are called “sinks”. The flow in sinks is reversed, flowing not out, but into the source. The mind boggles!

Here is the equation for the stream function of an axisymmetric line source $\Psi_2$

$$\Psi_2 = m(PO – PA)/a$$

It doesn’t get much better than this! We have:

$$PO = (x^2 + y^2)^{1/2}$$
$$PA = [(x-a)^2 + y^2]^{1/2}$$

Here, $m$ is the strength of the source, and $a$ is its length. Check the diagram below.

Now we have our free stream and line source stream functions, we are allowed to add them.

$$\Psi = \Psi_1 + \Psi_2$$
$$\quad = -Uy^2/2 + m(PO – Pa)/a$$
$$\quad = -Uy^2/2 + m[(x^2 + y^2)^{1/2} - (x-a)^2 + y^2]^{1/2}/a$$
As this is a linear theory, we could have written in some more stream functions, say n, viz:

\[ \Psi = \Psi_1 + \Psi_2 + \Psi_3 + \ldots + \Psi_n \]

Now this is really neat. With enough stream functions, we could create any shape we want! Well, maybe. Being mere mortals, we will end up using just 4 to get a good enough result for engineering purposes. I.e., within 20%. Don’t let anyone tell you engineering isn’t rough!

Just for the moment, we will get results for one line source only.

What we are now after are the velocity components at any point on any streamline. Let’s call these \( u \) on the abscissa and \( v \) on the ordinate (x and y axes respectively). Then a result from the theory is this:

\[ u = \frac{-\partial \Psi}{\partial y} / y \]
\[ v = \frac{-\partial \Psi}{\partial x} / y \]

Carrying out these partial derivatives yield:

\[ u = -U \cdot m(x^2+y^2)^{1/2} / a + m[(x-a)^2+y^2]^{1/2} / a \]
\[ v = mx(x^2+y^2)^{1/2} / (ay) - m(x-a)[(x-a)^2+y^2]^{1/2} / (ay) \]

Finally, it is useful to know the stagnation point. In the axisymmetric case, this occurs where \( u = 0 \) (reasonable, the stagnation point is defined as the point where there is no flow in the x-direction) and \( y = 0 \) (i.e., on the axis of rotation of the axisymmetric body).

Then from our equation for \( u \), with \( u = 0 \) and \( y = 0 \), we have:

\[ U = -m(x^2)^{1/2} / a + m[(x-a)^2]^{-1/2} / a \]
\[ = m/(x(x-a)) \]

If we have more than 1 line source, say \( n \), the relations below hold:

\[ u = u_1 + u_2 + \ldots + u_n \]
\[ v = v_1 + v_2 + \ldots + v_n \]

I leave it to the reader to prove that these stream functions are irrotational. The required condition is given below.

\[ \frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = 0 \]

Finally, I append my Quick Basic code for you fellow dinosaurs out there still running DOS, or even WIN98, which look better and better to me as each day of Windows Vista passes.
To recap, we set out to get the inflow field to a propeller, as affected by the presence of the spinner and cowl. This permits a correction to the blade angle, which is determined by the local axial velocity.

Regrettably, I need to point out some apparent errors elsewhere. In “Fundamentals of Fluid Mechanics”, Editions 1 and 2, by Gerhart and Gross, equations 9.36 and 9.37 are wrong. In “Teach Yourself Calculus”, by Abbott, page 122 (section 78), the differential symbols are missing for the subjects arcsin(x/a), arctan(x/a) and arcsec(x/a).

Appendix: Quick Basic source code for axisymmetric flow.

```basic
' Source in Uniform Stream Milne-Thompson p456
10/4/2009
' Axial symmetry 3-dimensional case ...
' set up arrays
DIM psi(30), x(10, 1200), y(10, 1200)  ' set up arrays
DIM r(2000), p(2000), xx(1200), yy(1200), Uu(20), Vv(20)
SCREEN 9, , 1, 1: pi = 4 * ATN(1): rtd = 180 / pi: dtr = pi / 180
scl = 220          ' centimetres across screen
WINDOW (-.1 * scl, -.3 * scl)-(.9 * scl, .3 * scl)

' Set input parameter values
U = 900         ' velocity of uniform stream (cm/second)
' (same thing as airspeed)
' spinner source parameters
m(1) = 7000       ' source strength (cubic
metres/stearadian)
a(1) = 5           ' length of line source (cm)
offset(1) = 2      ' offset of source from origin (cm)

' cowl source parameters
m(2) = 150000
a(2) = .1
offset(2) = 30

' pinch spinner/cowl source parameters
m(3) = -90000
a(3) = 30
offset(3) = 30

' rear fuselage source parameters
```
m(4) = -110000  
\(a(4) = 200\)  
offset(4) = 70

' select sources to be represented

\(\text{startp} = 1\) \ ' Start at source startp  
\(\text{stopp} = 4\) \ ' Finish at source stopp, in sequence

\(xprop = 20\) \ ' location of prop behind stagnation point  
\(\text{propdia} = 63.5\) \ ' cm

' find values of \(\psi\) (flux) for streamlines at \(-\infty\)  
' these streamlines are for the uniform flow with zero source strength  
' they are chosen to give even vertical spacing of the initial streamlines

dely = 5 \ ' spacing of initial streamlines in cm  
FOR n = 0 TO 10
\(\psi(n) = -.5 \times U \times (n \times \text{dely})^2\) \ ' \([3]\)'al flux for streamline \(n\)
NEXT n  
' \(n \times \text{dely}\) is height above fluid axis, cm

' plot free-stream streamlines

\(\text{FOR n} = 0 \text{ TO 12: LINE (-10, n \times \text{dely})-(30, n \times \text{dely}), 3:}\)

NEXT n

' draw line sources

g = 0  
FOR p = startp TO stopp  
g = g + .03: \text{COLOR 4: IF } m(p) < 0 \text{ THEN COLOR 2}
\text{LINE (offset(p), -g)-(a(p) + offset(p), g), , BF}\nNEXT p

' stagnation point

FOR x = -10 + .07 TO 20 STEP .1  
Uu = -U \ ' freestream velocity \(U\)  
\text{FOR p = startp TO stopp}
\text{GOSUB Uu} \ ' source axial velocities
Uu = Uu + uq \ ' total axial velocity
\text{NEXT p}
\text{IF Uu > 0 THEN GOTO ss} \ ' near enough to stagnation point
\text{NEXT x}

\text{ss: LINE (x, -.3)-(x, .3), 9}
xstag = x

' dividing streamline occurs at \(\psi = -Sm(p)\)

\(\text{psi} = 0\)  
\text{FOR p = startp TO stopp}
\text{psi = psi - m(p)}
\text{NEXT p}
\begin{verbatim}
j = 0
FOR x = xstag TO xstag + 260 STEP .25
    FOR y = 0 TO 25 STEP .25
        dum = psi + .5 * U * y ^ 2  \quad \text{\textit{test inequality value}}
        dump = 0
    FOR p = startp TO stopp
        GOSUB linesource
        dump = dump + psiline  \quad \text{\textit{line source stream function}}
    NEXT p
    IF dump < dum THEN GOTO 114
    NEXT y
114  j = j + 1: xx(j) = x: yy(j) = y
        LINE -(x, y)
    NEXT x

jj = j
FOR i = 2 TO jj
    IF x = xx(i - 1) THEN GOTO 115
    LINE (xx(i - 1), yy(i - 1))-(xx(i), yy(i)), 4
    LINE (xx(i - 1), -yy(i - 1))-(xx(i), -yy(i)), 4
    NEXT i
115  \quad \text{\textit{fill in nose shape}}

FOR k = 1 TO jj - 1
    IF x = xx(k - 1) THEN GOTO 117
    LINE (xx(k), yy(k))-(xx(k + 1), 0), 10, BF
    LINE (xx(k), -yy(k))-(xx(k + 1), 0), 10, BF
    NEXT k
117  \quad \text{\textit{draw source, origin}}

tt:
    LINE (0, -.1)-(a(p), .1), 4, BF  \quad \text{\textit{draw source}}
    LINE (-.3, -.3)-(.3, .3), 12  \quad \text{\textit{mark origin, x}}
= 0
    LINE (-.3, .3)-(.3, -.3), 12

\quad \text{\textit{draw prop}}

xp = xstag + xprop
    LINE (xp - 2, 0)-(xp - 1, propdia / 2), 9
    LINE (xp + 2, 0)-(xp + 1, propdia / 2), 9
    LINE (xp - 1, propdia / 2)-(xp + 1, propdia / 2), 9
    LINE (xp - 2, 0)-(xp + 2, 0), 9
    PAINT (xp, propdia / 4), 2, 9

    LINE (xp - 2, 0)-(xp - 1, -propdia / 2), 9
    LINE (xp + 2, 0)-(xp + 1, -propdia / 2), 9
    LINE (xp - 1, -propdia / 2)-(xp + 1, -propdia / 2), 9
    LINE (xp - 2, 0)-(xp + 2, 0), 9
    PAINT (xp, -propdia / 4), 2, 9

\quad \text{\textit{draw cm scale, centred at prop}}

    LINE (xp, -35)-(xp, 35), 12
FOR j = -35 TO 35 STEP 5
\end{verbatim}
' Get axial velocities u at cm stations, location x = xprop
x = xprop: i = 0
FOR y = 15 TO 30 STEP 5  ' cm stations along prop
   i = i + 1: yy(i) = y
dum1 = 0: dum2 = 0
FOR p = startp TO stopp
   GOSUB velocity
   dum1 = dum1 + uq: dum2 = dum2 + vq  ' line-source integraton for u
NEXT p
Uu = dum1 + U
Vv = dum2
UU(i) = UU  ' total axial velocity
Vv(i) = Vv  ' " radial "
Vstream(i) = SQR(Uu(i) ^ 2 + Vv(i) ^ 2)
NEXT y

' draw line source
   g = 0
FOR p = startp TO stopp
   g = g + .04: COLOR 4: IF m(p) < 0 THEN COLOR 2
   LINE (offset(p), -g)-(a(p) + offset(p), g), , BF
NEXT p
   k = i
FOR i = 1 TO k
   LOCATE i + 2, 36: COLOR 12
   PRINT USING "##  ####.#  ####.#  ####.#  ##.## "; yy(i); Uu(i); Vv(i); Vstream(i); Uu(i) / U
   NEXT i

   LOCATE 1, 36: COLOR 10: PRINT "cm Uaxial Vradial Wlocal Vratio"
   COLOR 2
   LOCATE 3, 5: PRINT USING " Prop diameter (mm) ###.#";
   propdia * 10
   LOCATE 4, 5: PRINT USING " Free stream vel (m/s) ###.#"; U / 10
   LOCATE 2, 5: COLOR 12: PRINT "Sea Fury"
   LOCATE 22, 8: PRINT "Axisymmetric representation of a Sea Fury fuselage"

' draw desired spinner
   xs(1) = 0: ys(1) = 0
   xs(2) = 4: ys(2) = 4
   xs(3) = 10: ys(3) = 7
   xs(4) = 20: ys(4) = 10
   FOR i = 2 TO 4
LINE (xs(i - 1) + xstag, ys(i - 1))-(xs(i) + xstag, ys(i))
LINE (xs(i - 1) + xstag, -ys(i - 1))-(xs(i) + xstag, -ys(i))
NEXT i

' mark cowling edges as circles
Rc = 6: Jc = xprop + 10: Kc = 35.6 / 2 - Rc: jj = -1:
GOSUB circ
Rc = 6: Jc = xprop + 10: Kc = -35.6 / 2 + Rc: jj = 1:
GOSUB circ

' compute stream lines for axial symmetry
nlines = 5
FOR n = 0 TO nlines STEP 1 ' each n is a new streamline
  k = 0: stepp = .4
  FOR x = 300 TO -100 STEP -stepp
    FOR y = .01 TO 10000 STEP stepp
      psi = -.5 * U * y ^ 2 ' free stream, axial symmetry
      FOR p = startp TO stopp
        GOSUB linesource
        psi = psi + psiline ' free stream + sources
      NEXT p
      IF psi > psi(n) THEN GOTO 5 ELSE GOTO 10
      5 NEXT y
    10 k = k + 1
      x(n, k) = x ' coordinates of streamline
      y(n, k) = y
    NEXT x
    endk = k - 1 ' counter for points on streamline
  NEXT n ' next streamline

' plot streamlines on screen
FOR n = 1 TO nlines
  FOR k = 2 TO endk
    COLOR 14: LINE (x(n, k - 1), y(n, (k - 1)))-(x(n, k), y(n, k))
    LINE (x(n, k - 1), -y(n, (k - 1)))-(x(n, k), -y(n, k))
  NEXT k
NEXT n
COLOR 0
END

linesource: ' line source stream function

  ' a = length of source
  ' m = strength of source

  ' po = SQR(x ^ 2 + y ^ 2)
  ' pa = SQR((x - a) ^ 2 + y ^ 2)
  ' psi = m / a * (po - pa)
mmq = m(p) / a(p)  ' line source strength
xq = x - offset(p)
po = SQR(xq ^ 2 + y ^ 2)
pa = SQR((xq - a(p)) ^ 2 + y ^ 2)
psiline = mmq * (po - pa)

RETURN

Uu:   ' compute axial velocity at distance x from source on axis
      ' for getting the stagnation point

mmq = m(p) / a(p)  ' individual source strength
xq = x - offset(p)
po = SQR(xq ^ 2 + y ^ 2)
pa = SQR((xq - a(p)) ^ 2 + y ^ 2)
xq = x - offset(p)
uq = -mmq / xq + mmq / (xq - a(p))

RETURN

velocity:  ' compute axial and transverse velocities
      ' at distance x,y from source

mmq = m(p) / a(p)  ' individual source strength
xq = x - offset(p)
po = SQR(xq ^ 2 + y ^ 2)
pa = SQR((xq - a(p)) ^ 2 + y ^ 2)
dum11 = 1 / SQR(xq ^ 2 + y ^ 2)
dum22 = 1 / SQR((xq - a(p)) ^ 2 + y ^ 2)
uq = -mmq * dum11 + mmq * dum22
vq = mmq * dum11 * xq / y - mmq * dum22 * (xq - a(p)) / y

RETURN

circ:  ' True circular-arc subroutine

'scl = .74
'xmin = -7: xmax = 7
'ymin = xmin * scl: ymax = xmax * scl
'WINDOW (xmin, ymin)-(xmax, ymax)
'COLOR 14
'  Rc = 5 / 2: Jc = 0: Kc = 0: jj = -1: GOSUB circ
'  jj = 1: GOSUB circ

IF Rc < .00001 THEN RETURN
xoff = 100: yoff = 100  ' offsets to avoid -ve sq. roots
xleft = xoff - Rc: xright = xoff + Rc  ' limits x
ylower = yoff - Rc: yupper = yoff + Rc  ' limits y
LINE (xleft - xoff + Jc, Kc)-(xleft - xoff + Jc, Kc)

FOR ec = xleft TO xright STEP (xright - xleft) / 100
IF ABS(ec - xoff) > ABS(Rc) THEN 2221
yyc = SQR(Rc ^ 2 - (ec - xoff) ^ 2)
xxc = ec - xoff
LINE -(xxc + Jc, -yyc * jj + Kc)
NEXT ec
LINE -(xright - xoff + Jc, Kc)
RETURN