Subsonic thin profile - visualization

March 28, 2025

[1]:
```
import numpy              as np
import matplotlib.pyplot as plt
```

```
/home/lukas/.local/lib/python3.10/site-
packages/matplotlib/projections/__init__.py:63: UserWarning: Unable to import
Axes3D. This may be due to multiple versions of Matplotlib being installed (e.g.
as a system package and as a pip package). As a result, the 3D projection is not
available.
  warnings.warn("Unable to import Axes3D. This may be due to multiple versions
of "
```

1 Parameters

[2]:
```
M       = 0.7                    # Mach number
epsilon = 0.1                    # Profile thickness
H       = lambda x: 0.5 - 2*x**2  # normalized thickness function
dHdx    = lambda x: -4*x          # H'
```

2 Prandtl-Glauert transformation

[3]:
```
beta    = np.sqrt(1-M**2)
H_t     = lambda x: beta * H(x)
dHdx_t = lambda x: beta * dHdx(x)
```

3 Analytical incompressible flow

The equivalent source/sink distribution $q(x)$ along the axis of the profile is defined by the transformed asymptotic flow-tangency condition (6.3) as follows:

$$
\begin{aligned}
q(x) &= \bar{\phi}_{\bar{Y}}(X, 0^+) - \bar{\phi}_{\bar{Y}}(X, 0^-) \\
&= 2\bar{\phi}_{\bar{Y}}(X, 0^+) \\
&= 2\beta H'.
\end{aligned}
$$

The analytical solution for an incompressible potential flow driven by a continuous source sheet can be found, e.g., in Schneider (1978), p. 122:

$$\bar{\phi} = \frac{1}{2\pi} \int\limits_{-1/2}^{1/2} q(\xi) \ln \sqrt{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi.$$

Substituting the expression for $q(x)$ we obtain

$$\bar{\phi} = \frac{\beta}{\pi} \int\limits_{-1/2}^{1/2} H'(\xi) \ln \sqrt{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi.$$

If we consider again the profile shape described by the quadratic function \$ H(x) = 1/2 - 2X^2\$, then $H' = -4X$ and

$$\bar{\phi} = \frac{-4\beta}{\pi} \int\limits_{-1/2}^{1/2} \xi \ln \sqrt{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi.$$

The analytical solution of the integral (found with Wolfram Mathematica) reads

$$\bar{\phi} = \frac{2\beta}{\pi} \left[X - 2X\bar{Y} \left(\arctan \frac{1/2 - X}{\bar{Y}} + \arctan \frac{1/2 + X}{\bar{Y}} \right) + (1/4 - X^2 + \bar{Y}^2) \arctan \frac{4X}{1 + 4X^2 + 4\bar{Y}^2} \right].$$

More importantly, we are interested in the velocity components, that are defined as follows:

$$\begin{aligned} \bar{U} = \frac{\partial \bar{\phi}}{\partial X} &= \frac{1}{2\pi} \int\limits_{-1/2}^{1/2} q(\xi) \frac{X-\xi}{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi \\ &= \frac{\beta}{\pi} \int\limits_{-1/2}^{1/2} H'(\xi) \frac{X-\xi}{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi \\ &= \frac{-4\beta}{\pi} \int\limits_{-1/2}^{1/2} \xi \frac{X-\xi}{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi \end{aligned}$$

and

$$\bar{V} = \frac{\partial \bar{\phi}}{\partial \bar{Y}} = \frac{1}{2\pi} \int_{-1/2}^{1/2} q(\xi) \frac{\bar{Y}}{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi$$

$$= \frac{\beta}{\pi} \int_{-1/2}^{1/2} H'(\xi) \frac{\bar{Y}}{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi$$

$$= \frac{-4\beta}{\pi} \int_{-1/2}^{1/2} \xi \frac{\bar{Y}}{(X-\xi)^2 + \bar{Y}^2} \mathrm{d}\xi.$$

Again, there exist analytical solutions of the integrals, and the velocity components can be expressed in closed form as follows:

$$\bar{U} = \frac{4\beta}{\pi} \left[1 - \bar{Y} \left(\arctan \frac{1/2 - X}{\bar{Y}} + \arctan \frac{1/2 + X}{\bar{Y}} \right) - X \operatorname{arctanh} \frac{4X}{1 + 4X^2 + 4\bar{Y}^2} \right]$$

$$\bar{V} = \frac{-4\beta}{\pi} \left[X \left(\operatorname{arccot} \frac{2\bar{Y}}{1-2X} + \arctan \frac{1/2 + X}{\bar{Y}} \right) + \frac{\bar{Y}}{2} \ln \left(1 - \frac{8X}{(1+2X)^2 + 4\bar{Y}^2} \right) \right]$$

[4]:
```
u_bar = lambda x,y_bar: 4*beta/np.pi * ( 1
                                    -y_bar*( np.atan((0.5-x)/y_bar)
                                            +np.atan((0.5+x)/y_bar)
                                           )

                                       -x*np.atanh( 4*x/(
                                              1+4*x**2+4*y_bar**2
                                                         )
                                                  )
                                          )

v_bar = lambda x,y_bar: -4*beta/np.pi *( x*( np.atan((1-2*x)/(2*y_bar))
                                           +np.atan((1+2*x)/(2*y_bar))
                                          )
                                        +y_bar/2 *np.log( 1 - 8*x/(
                                                               (1+2*x)**2
                                                              +4*y_bar**2
                                                                    )
                                                         )
                                          )
```

Thus, we can plot the auxiliary incompressible flow over the transformed profile defined by $\bar{Y} = \pm\epsilon\beta H(X)$.

3.1 Visualization

[5]:
```
xmin, xmax = -1.0 , 1.0  # boundaries of the visualized region
ymin, ymax = -0.5 , 0.5
N          = 10          # number of velocity vectors in each direction
```

[6]:
```
x,y = np.meshgrid( np.linspace(xmin,xmax,N) ,
                   np.linspace(ymin,ymax,N) )
y_bar = beta * y
```

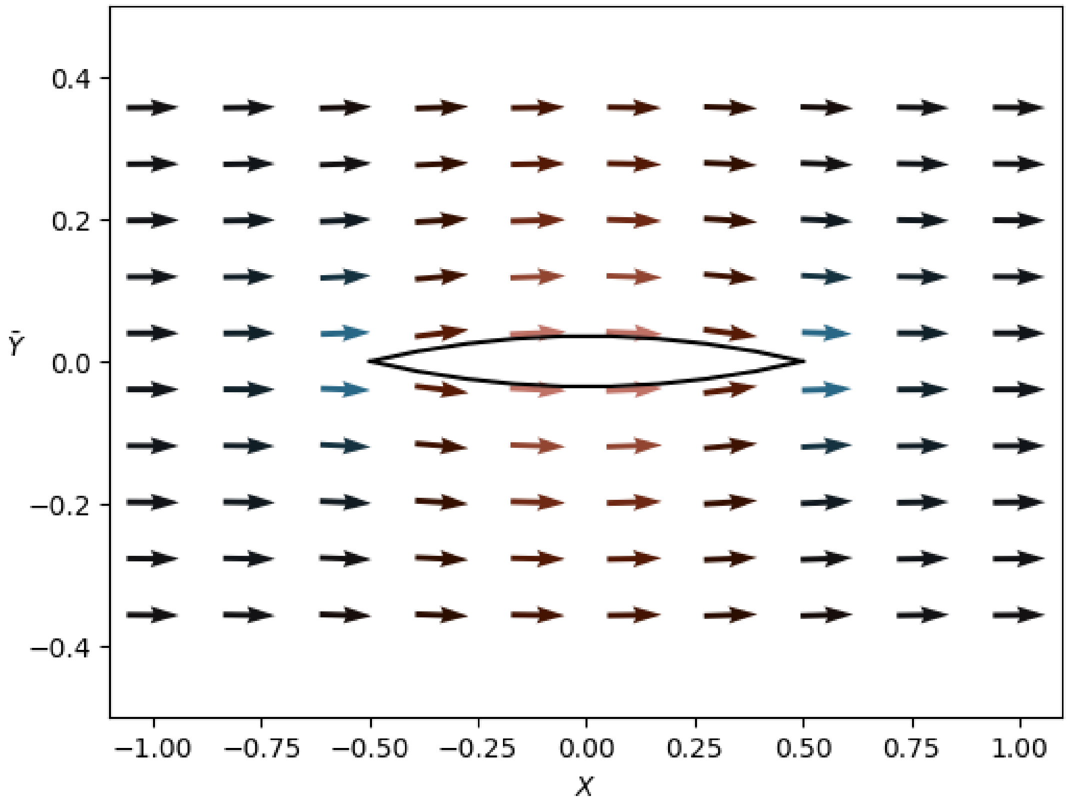
[7]:
```
u_ = 1 + epsilon * u_bar(x,y_bar)
v_ =     epsilon * v_bar(x,y_bar)
```

[8]:
```
# Velocity vectors
plt.quiver(x,y_bar,u_,v_,np.sqrt(u_**2+v_**2),angles='xy',
           pivot='mid',clim=(1-epsilon,1+epsilon),cmap='berlin')

# Shape of the profile
x_shape     = np.linspace(-0.5, 0.5, N)
y_shape_bar = epsilon * beta * H(x_shape)
plt.plot(x_shape, y_shape_bar, 'k-', x_shape, -y_shape_bar, 'k-')

# Labels
plt.xlabel(r'$X$')
plt.ylabel(r'$\bar{Y}$',rotation='horizontal')
plt.ylim(ymin, ymax)
```

[8]:
```
(-0.5, 0.5)
```



4 Compressible flow

The subsonic compressible flow is obtained by backward transformation into the original variables

$$Y = \bar{Y}/\beta$$
$$U_1 = \bar{U}/\beta^2$$
$$V_1 = \bar{V}/\beta$$

[9]:
```
u1 = lambda x,y: u_bar(x,y*beta) / beta**2
v1 = lambda x,y: v_bar(x,y*beta) / beta
```

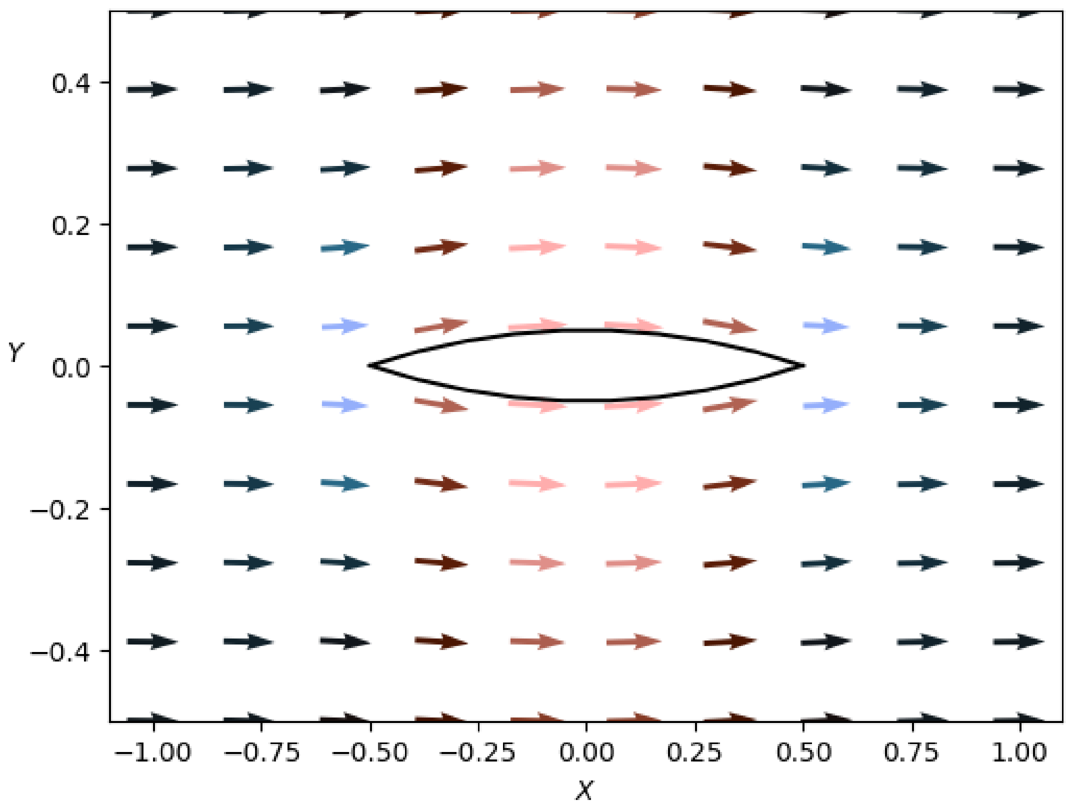
[10]:
```
u  = 1 + epsilon * u1(x,y)
v  =     epsilon * v1(x,y)

y_shape = y_shape_bar / beta
```

[11]:
```
plt.quiver(x,y,u,v,np.sqrt(u**2+v**2),angles='xy',
           pivot='mid',clim=(1-epsilon,1+epsilon),cmap='berlin')
plt.plot(x_shape, y_shape, 'k-', x_shape, -y_shape, 'k-')
```

```
# Labels
plt.xlabel(r'$X$')
plt.ylabel(r'$Y$',rotation='horizontal')
plt.ylim(ymin, ymax)
```

[11]: (-0.5, 0.5)



[ ]: