Subsonic thin profile - streamlines

March 28, 2025

[1]:
```
import numpy               as np
import matplotlib.pyplot   as plt
from   scipy.integrate import solve_ivp
```

```
/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       = 0.8                       # 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)
```

3 Auxiliary 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

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

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

$$
\begin{aligned}
\bar{U} &= \frac{4\beta}{\pi}\left[1-\bar{Y}\left(\arctan\frac{1/2-X}{\bar{Y}}+\arctan\frac{1/2+X}{\bar{Y}}\right)-X\,\mathrm{arctanh}\frac{4X}{1+4X^2+4\bar{Y}^2}\right] \\
\bar{V} &= \frac{-4\beta}{\pi}\left[X\left(\mathrm{arccot}\frac{2\bar{Y}}{1-2X}+\arctan\frac{1/2+X}{\bar{Y}}\right)+\frac{\bar{Y}}{2}\ln\left(1-\frac{8X}{(1+2X)^2+4\bar{Y}^2}\right)\right]
\end{aligned}
$$

[4]:
```
u_bar = lambda x,y_bar,beta: 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,beta: -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 $\bar{Y} = \pm\epsilon\beta H(X)$.

[5]:
```
u_ = lambda x,y_bar:  1 + epsilon * u_bar(x,y_bar,beta)
v_ = lambda x,y_bar:      epsilon * v_bar(x,y_bar,beta)
```

The pressure perturbation $(p - p_\infty)$ is typically expressed in dimensionless form as the *pressure coefficient*

$$C_p = \frac{p - p_\infty}{\rho_\infty u_\infty^2 / 2}.$$

3.1 Visualization

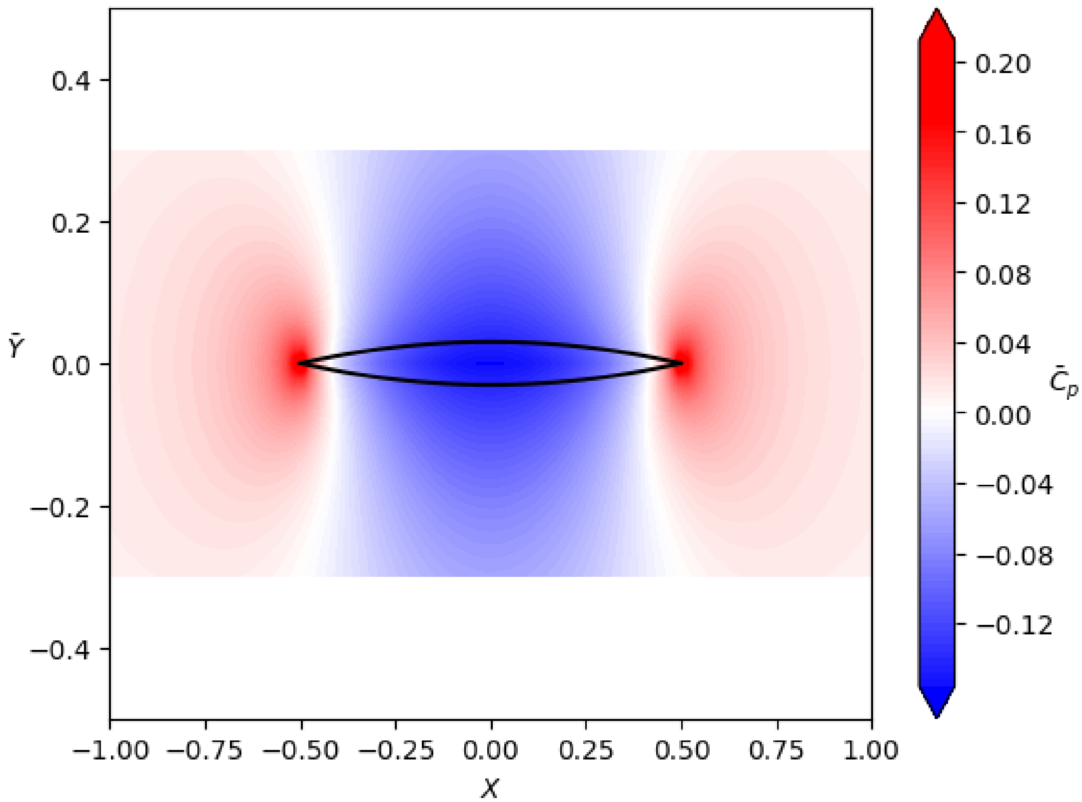
[6]:
```
xmin, xmax = -1.0 , 1.0  # boundaries of the visualized region
ymin, ymax = -0.5 , 0.5
N          = 100         # resolution
```

[7]:
```
x,y = np.meshgrid( np.linspace(xmin,xmax,N) ,
                   np.linspace(ymin,ymax,N) )
y_bar = beta * y
```

[8]:
```
u_grid_ = u_(x,y_bar)
v_grid_ = v_(x,y_bar)
u_mag_ = np.sqrt(u_grid_**2+v_grid_**2)
p_ = 1 - u_mag_**2
```

[9]:
```
fig,ax = plt.subplots()

# Pressure field
p_plot = ax.contourf(x,y_bar,p_,N,cmap='bwr',
             vmin=-epsilon/beta,vmax=epsilon/beta,extend='both')

cbar = plt.colorbar(p_plot)
cbar.set_label(r'$\bar{C}_p$'
              , rotation='horizontal')

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

# Labels
ax.set_xlabel(r'$X$')
ax.set_ylabel(r'$\bar{Y}$',rotation='horizontal')
ax.set_ylim(ymin, ymax)
```

[9]:
```
(-0.5, 0.5)
```

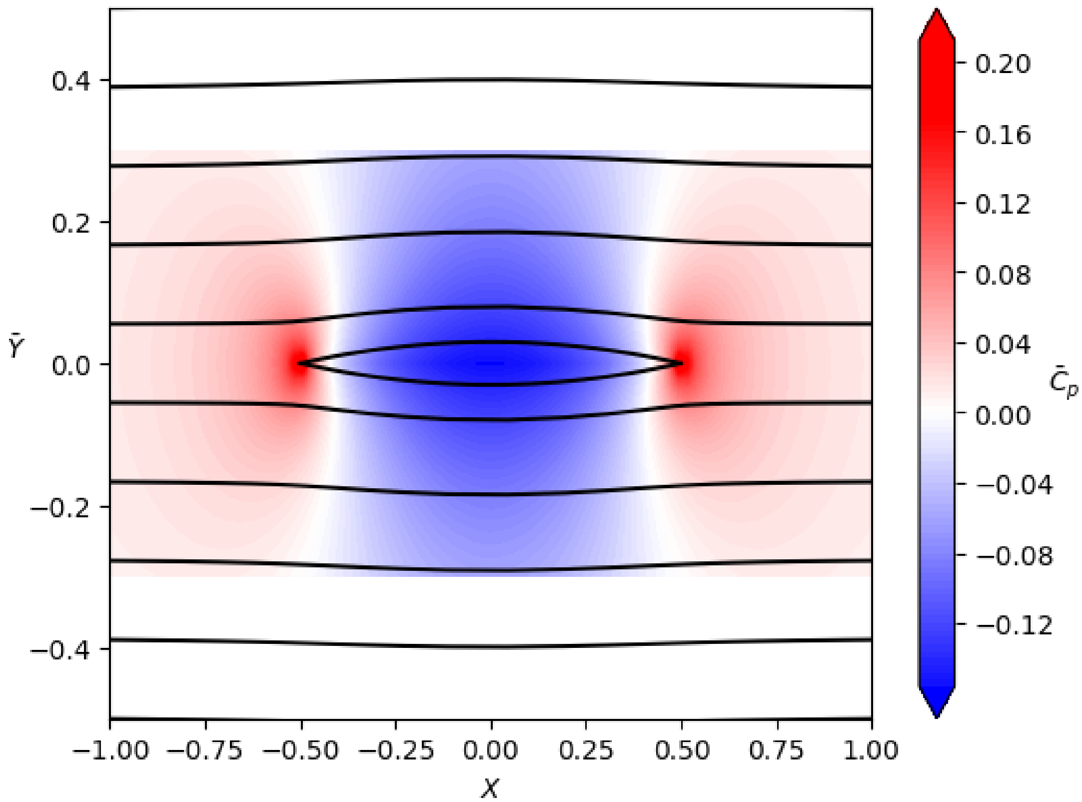


3.2 Streamlines

[10]:
```
dy_bar_dx = lambda x,y_: v_(x,y_) / u_(x,y_)

x_span = (xmin, xmax)
Ns = 10
y0 = np.linspace(ymin, ymax, Ns)

sol = solve_ivp(dy_bar_dx, x_span, y0, vectorized=True, atol=1e-8, rtol=1e-8)
```

[11]:
```
ax.plot(sol.t, sol.y.T, 'k-')
fig
```

[11]:



4 Compressible flow

4.1 Backward transformation

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

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

[12]:
```
y_shape = y_shape_bar / beta
```

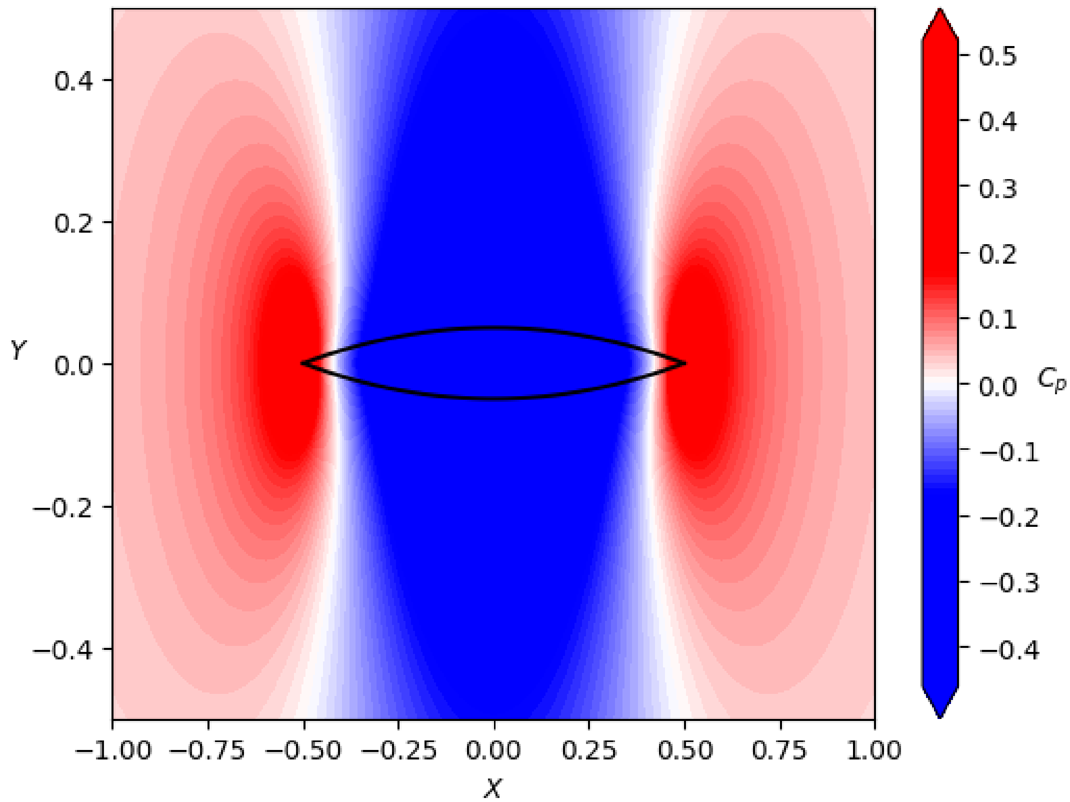
[13]:
```
u1 = lambda x,y,beta: u_bar(x,y*beta,beta) / beta**2
v1 = lambda x,y,beta: v_bar(x,y*beta,beta) / beta

u  = lambda x,y,beta: 1 + epsilon * u1(x,y,beta)
v  = lambda x,y,beta:     epsilon * v1(x,y,beta)
```

4.2 Velocity and pressure fields

[14]:
```
u_grid = u(x,y,beta)
v_grid = v(x,y,beta)
u_mag  = np.sqrt( u_grid**2 + v_grid**2 )
p      = 1-u_mag**2
```

[15]:
```
fig,ax = plt.subplots()

# Pressure field
p_plot = ax.contourf(x,y,p,N,cmap='bwr',extend='both',
            vmin=-epsilon/beta,vmax=epsilon/beta)

cbar = plt.colorbar(p_plot)
cbar.set_label(r'$C_p$'
              , rotation='horizontal')

# Shape of the profile
ax.plot(x_shape, y_shape, 'k-', x_shape, -y_shape, 'k-')

# Labels
ax.set_xlabel(r'$X$')
ax.set_ylabel(r'$Y$',rotation='horizontal')
ax.set_ylim(ymin, ymax)
```

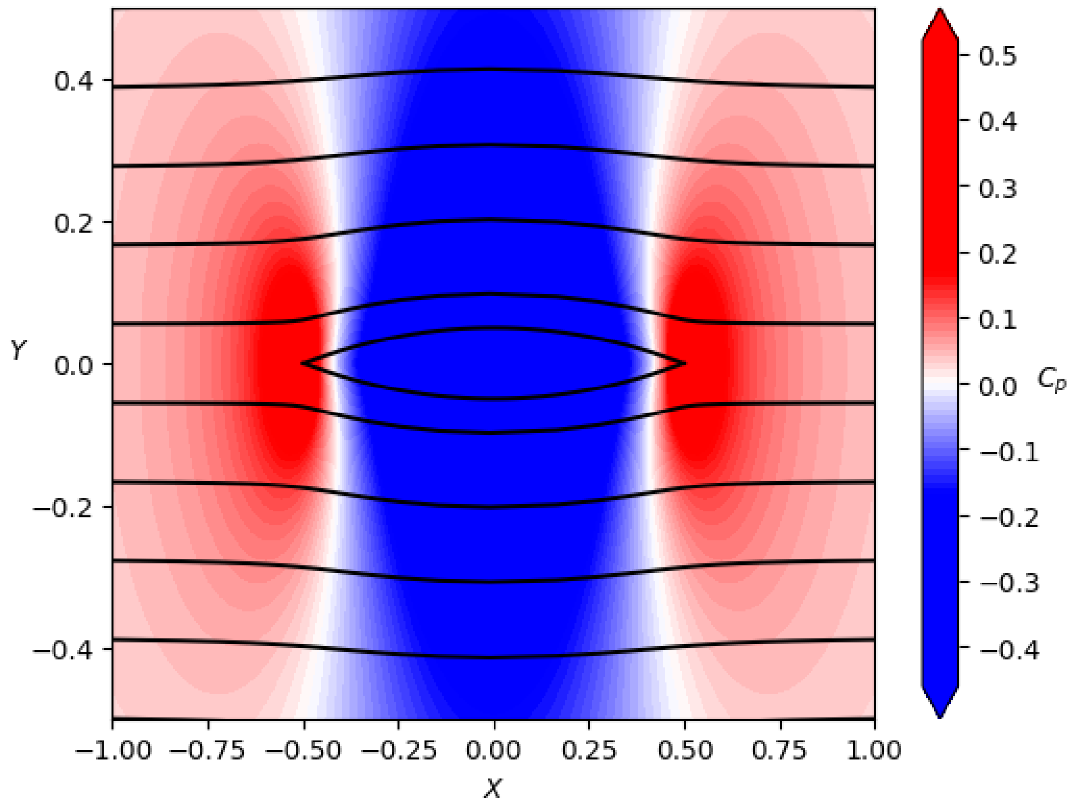
[15]:
```
(-0.5, 0.5)
```



4.3 Streamlines

[16]:
```
dydx = lambda x,y,beta: v(x,y,beta)/u(x,y,beta)
sol  = solve_ivp(dydx, x_span, y0, vectorized=True,
                 args=(beta,), atol=1e-8, rtol=1e-8)

ax.plot(sol.t, sol.y.T, 'k-')
fig
```

[16]:



5 Incompressible flow over the same profile

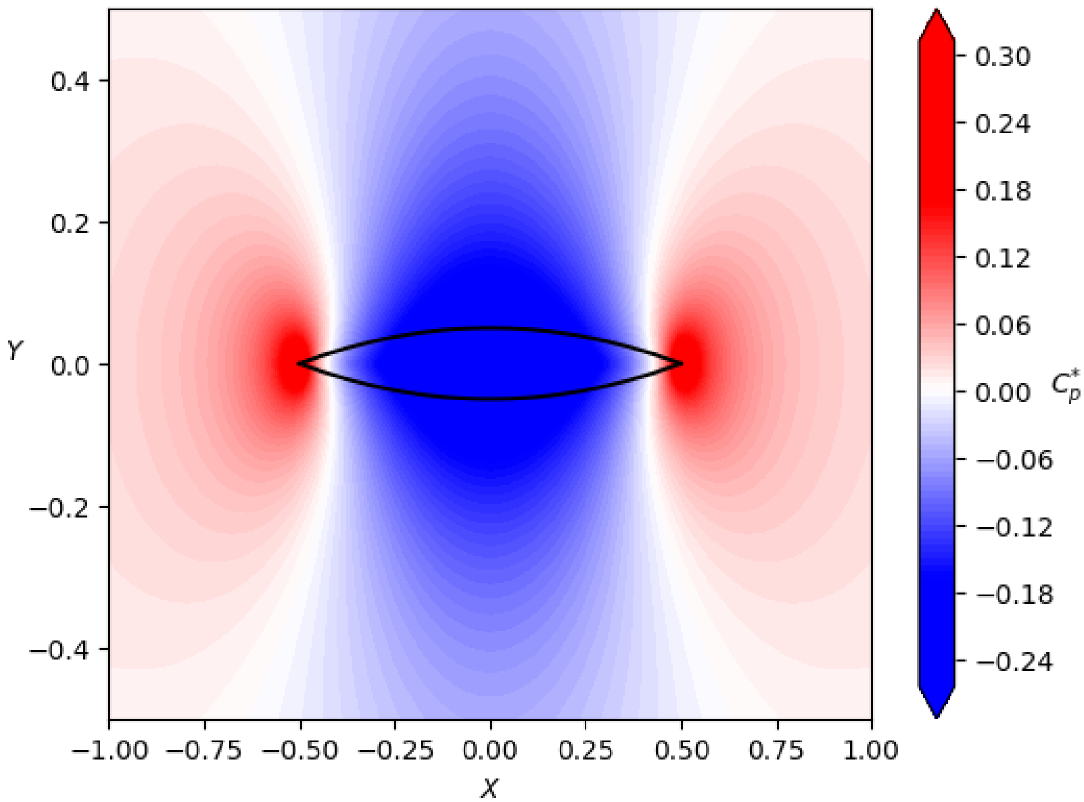
If the Mach number is small, $M_\infty^2 \ll 1$, then the flow can be assumed incompressible. The solution for an incompressible flow over the same profile is obtained by setting $M_\infty = 0$, such that $\beta = 1$.

[17]:
```
beta_inc = 1
u_inc = u(x,y,beta_inc)
v_inc = v(x,y,beta_inc)
u_mag_inc = np.sqrt( u_inc**2 + v_inc**2 )
p_inc = 1-u_mag_inc**2
```

[18]:
```
fig,ax = plt.subplots()

# Pressure field
p_plot = ax.contourf(x,y,p_inc,N,cmap='bwr',extend='both',
            vmin=-epsilon/beta,vmax=epsilon/beta)

cbar = plt.colorbar(p_plot)
cbar.set_label(r'$C_p^{*}$'
               , rotation='horizontal')
```

```python
# Shape of the profile
ax.plot(x_shape, y_shape, 'k-', x_shape, -y_shape, 'k-')

# Labels
ax.set_xlabel(r'$X$')
ax.set_ylabel(r'$Y$',rotation='horizontal')
ax.set_ylim(ymin, ymax)
```

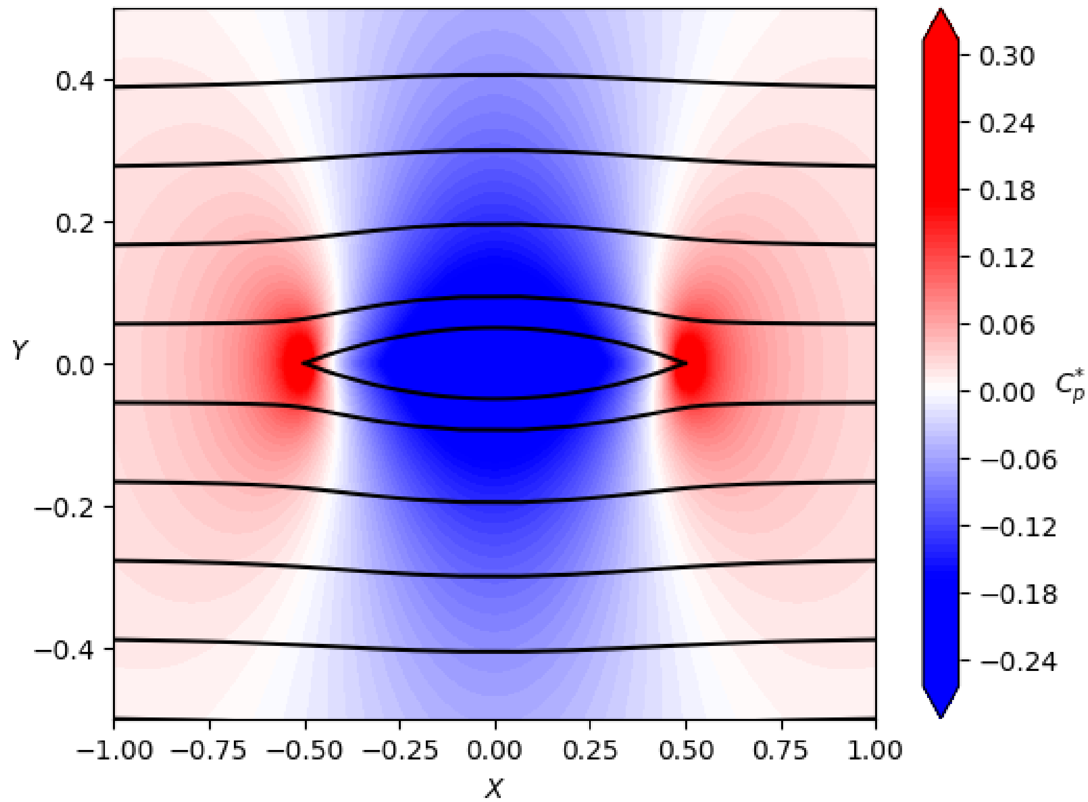
[18]: (-0.5, 0.5)



5.1 Streamlines

[19]:
```python
sol_inc = solve_ivp(dydx, x_span, y0, vectorized=True,
                    args=(beta_inc,), atol=1e-8, rtol=1e-8)

ax.plot(sol_inc.t, sol_inc.y.T, 'k-')
fig
```

[19]:



6 Comparison between incompressible and subsonic flows

The compressibility of the fluid (air) influences the magnitude of the pressure perturbation, which in the subsonic flow increases with increasing Mach number. For thin airfoils, the pressure coefficient C_p in the subsonic flow can be related to the pressure coefficient of an incompressible flow over the same profile (C_p^*) as follows:

$$C_p(X,Y) \approx \frac{C_p^*(X,\beta Y)}{\beta}.$$

[20]:
```
fig,ax = plt.subplots(nrows=1, ncols=2)

ax[0].contourf(x,y,p,N,cmap='bwr',extend='both',
            vmin=-epsilon/beta,vmax=epsilon/beta)

ax[0].set_ylabel(r'$Y$', rotation='horizontal')
ax[0].set_xlabel(r'$X$')
ax[0].set_title(r'$C_p (X,Y)$')

p_plot = ax[1].contourf(x,y/beta,p_inc/beta,N,cmap='bwr',extend='both',
```

```
             vmin=-epsilon/beta,vmax=epsilon/beta)

ax[1].set_ylim(ymin,ymax)
ax[1].set_xlabel(r'$X$')
ax[1].set_title(r'$C_p^{*}(X,\beta Y)/\beta$')

cbar = plt.colorbar(p_plot)
cbar.set_label(r'$C_p$'
               , rotation='horizontal')
```