# supersonic_thin_visualization

March 18, 2025

## 1 Visualization of the analytical solution for the supersonic flow over a symmetric thin profile

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

### 1.1 Parameters

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

### 1.2 Distribution of the velocity vectors

[3]:
```
xmin, xmax = -0.75, 1.5  # boundaries of the visualized region
ymin, ymax = -0.5 , 0.5
n          = 10          # number of vectors in each direction
```

Create a uniform grid of points (origins of the velocity vectors)

[4]:
```
x,y = np.meshgrid( np.linspace(xmin,xmax,n) ,
                   np.linspace(ymin,ymax,n) )
```

and transform the Cartesian coordinates to $\xi$ and $\eta$

[5]:
```
xi  = x - np.sqrt(M**2-1)*y
eta = x + np.sqrt(M**2-1)*y
```

### 1.3 Division into sub-regions

[6]:
```
upper = (y>0) & (xi >-1/2) & (xi <1/2)
lower = (y<0) & (eta>-1/2) & (eta<1/2)
```

### 1.4 Evaluate analytical velocity perturbation

The velocity components are given by

$$u = 1 + \epsilon \frac{\partial \phi_1}{\partial x},$$

$$v = \epsilon \frac{\partial \phi_1}{\partial y}.$$

We will denote

$$u_1 = \frac{\partial \phi_1}{\partial x} \quad \text{and} \quad v_1 = \frac{\partial \phi_1}{\partial y}.$$

We can express $u_1$ and $v_1$ in terms of the analytical solution $\tilde{\phi}$ as follows:

$$\frac{\partial \phi_1}{\partial x} = \begin{cases} \mathrm{d}\tilde{\phi}/\mathrm{d}\xi = -(M_\infty^2 - 1)^{-1/2} H'(\xi) & \text{for } y > 0 \wedge -1/2 < \xi < 1/2 \\ \mathrm{d}\tilde{\phi}/\mathrm{d}\eta = -(M_\infty^2 - 1)^{-1/2} H'(\eta) & \text{for } y < 0 \wedge -1/2 < \eta < 1/2 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\frac{\partial \phi_1}{\partial y} = \begin{cases} -(M_\infty^2 - 1)^{1/2} \mathrm{d}\tilde{\phi}/\mathrm{d}\xi = H'(\xi) & \text{for } y > 0 \wedge -1/2 < \xi < 1/2 \\ (M_\infty^2 - 1)^{1/2} \mathrm{d}\tilde{\phi}/\mathrm{d}\eta = -H'(\eta) & \text{for } y < 0 \wedge -1/2 < \eta < 1/2 \\ 0 & \text{otherwise} \end{cases}.$$

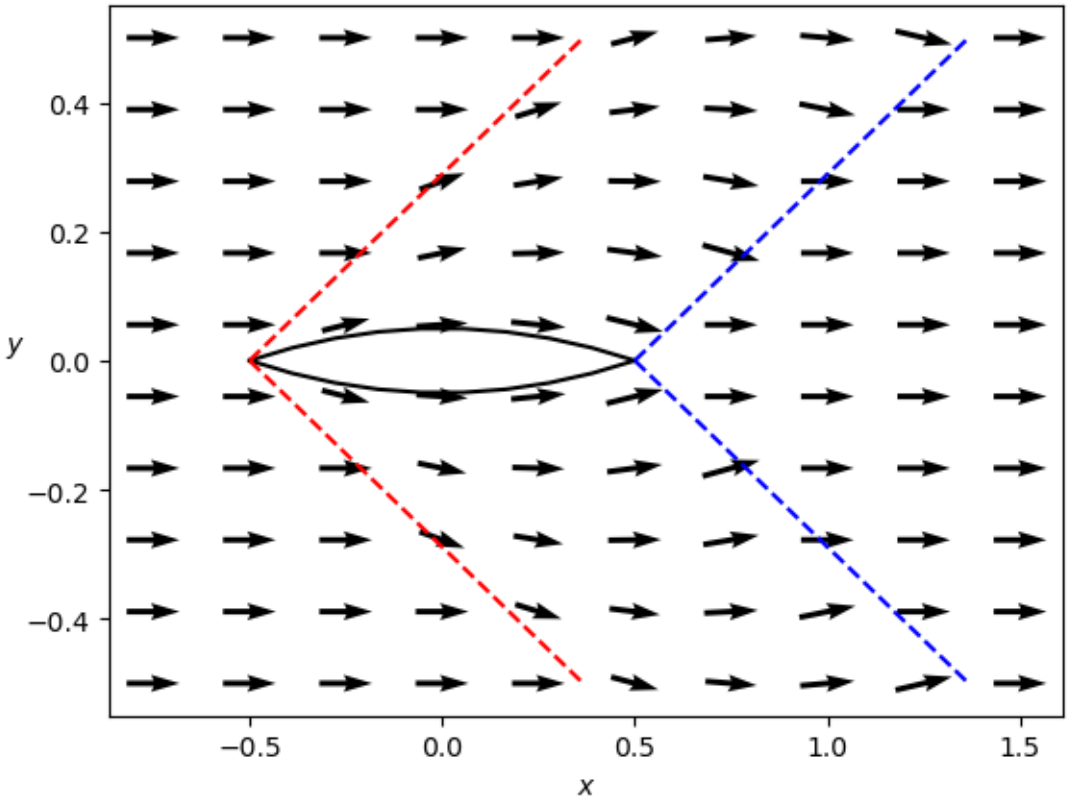
[7]:
```
u1 = np.zeros((n,n))
v1 = np.zeros((n,n))

u1[upper] = -dHdx( xi[upper]) / np.sqrt(M**2-1)
u1[lower] = -dHdx(eta[lower]) / np.sqrt(M**2-1)

v1[upper] =  dHdx( xi[upper])
v1[lower] = -dHdx(eta[lower])

u = 1 + epsilon * u1
v =     epsilon * v1
```

### 1.5 Plot

[8]:
```
# Pressure
'''
p = 0.5 - (u**2+v**2)/2
p_field = plt.contourf(x,y,p,np.linspace(-0.5,0.5,50),cmap='seismic',vmin=-0.
↪5,vmax=0.5)
cbar = plt.colorbar(p_field)
'''

# Velocity vectors
plt.quiver(x,y,u,v,angles='xy',pivot='mid')
```

```
# Shape of the profile
x_shape = np.linspace(-0.5, 0.5, n)
y_shape = epsilon * H(x_shape)
plt.plot(x_shape, y_shape, 'k-', x_shape, -y_shape, 'k-')

# Characteristics (shock-wave fronts)
y_char = np.array([0, ymax])
x_char = np.sqrt(M**2-1)*y_char
plt.plot(x_char-0.5, y_char, 'r--')
plt.plot(x_char-0.5,-y_char, 'r--')
plt.plot(x_char+0.5, y_char, 'b--')
plt.plot(x_char+0.5,-y_char, 'b--')

# Labels
plt.xlabel(r'$x$')
plt.ylabel(r'$y$',rotation='horizontal')
```

[8]: Text(0, 0.5, '$y$')




[ ]: