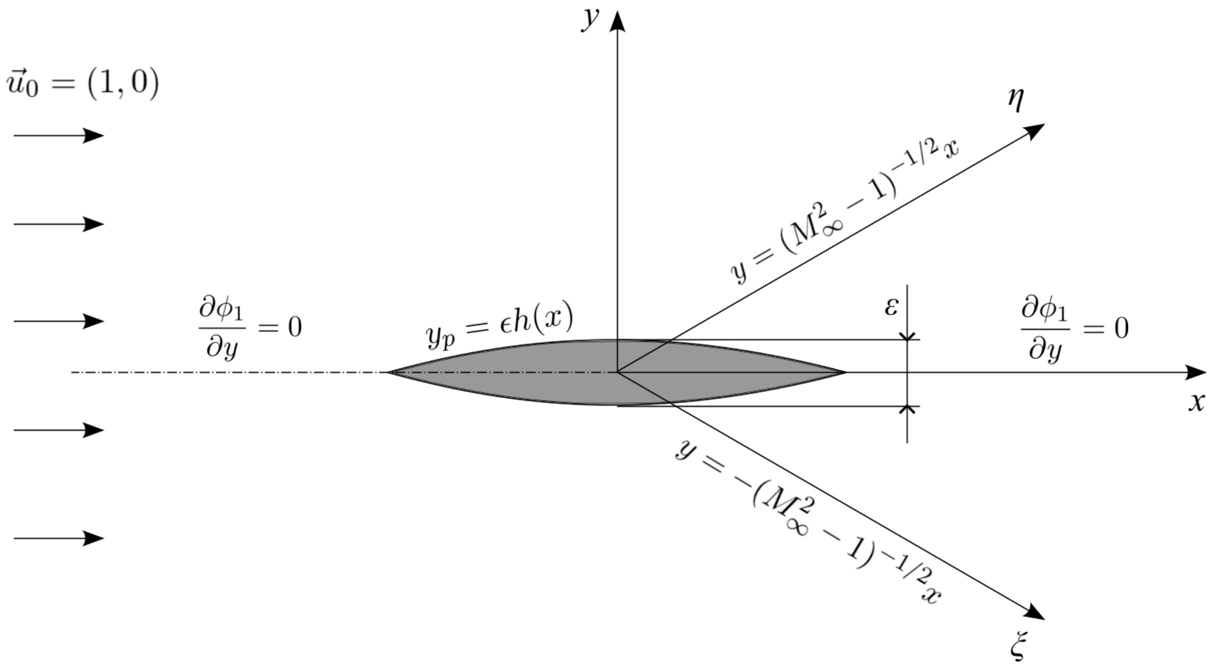
Supersonic thin profile

In this example we will compute the supersonic potential flow around a thin symmetric airfoil aligned with a horizontal incoming flow $\vec{u_0} = (1, 0)$.



Initialization

```
clearvars; % Clear workspace variables
close all; % Close all figure windows
```

We will employ symbolic variables for easier substitutions and computation of derivatives

```
syms x y M epsilon
assume(x       ,'real');
assume(y       ,'real');
assume(M       ,'real'); % Mach number
assumeAlso( M > 1 );
assume(epsilon,'real'); % Small thickness parameter
assumeAlso( epsilon>0 );
```

Shape

Consider a thin parabolic airfoil profile

$$y_p = \varepsilon\, h(x)$$

$$h(x) = 1 - x^2 \quad \text{for } -1 < x < 1$$

where $\varepsilon << 1$

```
h   = 1-x^2;     % local thickness of the airfoil
h_x = diff(h,x)  % derivative of h(x)
```

h_x = $-2\,x$

Potential flow

For large Reynolds numbers the flow is inviscid and irrotational

$$\nabla \times \vec{u} = \frac{\partial}{\partial x} v - \frac{\partial}{\partial y} u = 0$$

except from a thin boundary layer. For an irrotational flow there exists a velocity potential ϕ such that

$$u = \frac{\partial}{\partial x}\phi \text{ and } v = \frac{\partial}{\partial y}\phi .$$

Asymptotic expansion

We assume that the thin airfoil will cause a small perturbation of the flow, proportional to its thickness. Thus, we expand ϕ into power series for small values of ε:

$$\phi = \phi_0 + \varepsilon\,\phi_1$$

where $\phi_0 = x$

```
phi_0 = x;
```

represents the undisturbed incoming flow and ϕ_1 is the perturbation which is governed by the wave equation

$$(M_\infty^2 - 1)\phi_{1,xx} - \phi_{1,yy} = 0$$

with the boundary conditions

$$\begin{aligned} \phi_{1,y} &= 0 && \text{at } \{x \le -1 \cup x \ge 1\} \wedge y = 0 \\ \phi_{1,y} &= \pm h'(x) = \mp 2x && \text{at} \quad -1 < x < 1 \wedge y = 0^{\pm}, \\ \phi_1 &= 0 && \text{for} \quad x \to -\infty . \end{aligned}$$

Analytical solution

We transform the problem into a new coordinate system aligned with the characteristics

$$\xi = x - \sqrt{M_\infty^2 - 1}\, y,\ \eta = x + \sqrt{M_\infty^2 - 1}\, y$$

```
xi = x - sqrt(M.^2-1).*y ;
eta= x + sqrt(M.^2-1).*y ;
```

and search for a solution in the new coordinate system such that

$$\phi_1(x, y) = \tilde{\phi}_1(\xi(x, y), \eta(x, y)) .$$

The transformed wave equation reads

$$\frac{\partial^2 \tilde{\phi}_1}{\partial \xi \partial \eta} = 0 .$$

The analytical solution of the transformed wave equation takes the following form

$$\tilde{\phi}_1 = f(\xi) + g(\eta),$$

which can be transformed back to the original coordinate system as

$$\phi_1 = f\left(x - \sqrt{M_\infty^2 - 1}\, y\right) + g\left(x + \sqrt{M_\infty^2 - 1}\, y\right).$$

The functions *f* and *g* are determined by the boundary conditions.

Matching with the boundary conditions

Since boundary conditions are prescribed along the symmetry plane $y = 0$, we must split the domain into an upper side (*y* > 0) and a lower side (y < 0). The solutions in the upper and the lower side are reflection symmetric about *y* = 0.

Far-field condition

First we employ the far-field condition

$$\phi_1 \to 0 \quad \text{as } x \to -\infty$$

which in the new coordinate system transforms to

$$\tilde{\phi}_1 \to 0 \quad \text{as } \begin{cases} \xi \to -\infty & \text{for } y > 0 \\ \eta \to -\infty & \text{for } y < 0 \end{cases}.$$

Therefore,

$$\begin{cases} g(\eta) = \text{const.} = -f(-\infty) & \text{for } y > 0 \\ f(\xi) = \text{const.} = -g(-\infty) & \text{for } y < 0 \end{cases}.$$

We can conveniently set the constants to 0 such that

$$\tilde{\phi}_1 = \begin{cases} f(\xi) & \text{for } y > 0 \\ g(\eta) & \text{for } y < 0 \end{cases}$$

and

$$f(-\infty) \to 0, \quad g(-\infty) \to 0.$$

Symmetry condition

For the symmetry condition

$$\phi_{1,y} = 0 \quad \text{at } \{x \le -1 \cup x \ge 1\} \wedge y = 0$$

we transform the *y*-derivative on the left-hand side into the new coordinate system using the chain rule

$$\frac{\partial}{\partial y}\phi_1 = \begin{cases} \dfrac{\partial \xi}{\partial y}\dfrac{\mathrm{d}\, f}{\mathrm{d}\, \xi} = -\sqrt{M_\infty^2 - 1}\, f' & \text{for } y > 0 \\ \dfrac{\partial \eta}{\partial y}\dfrac{\mathrm{d}\, g}{\mathrm{d}\, \eta} = \sqrt{M_\infty^2 - 1}\, g' & \text{for } y < 0 \end{cases}.$$

Again, we have to define the boundary in terms of the new coordinates ξ and η. The leading and the trailing edges of the airfoil are located, respectively, at

$$(x, y) = (\mp 1, 0) \Longrightarrow (\xi, \eta) = (\mp 1, \mp 1).$$

The transformed symmetry condition reads

$$\begin{cases} f' = 0 & \text{at } \xi < -1 \cup \xi > 1 \text{ for } y > 0 \\ g' = 0 & \text{at } \eta < -1 \cup \eta > 1 \text{ for } y < 0 \end{cases}.$$

Integrating the symmetry condition we obtain

$$f(\xi < -1) = a,\; f(\xi > 1) = b,\; g(\eta < -1) = c,\; g(\eta > 1) = d.$$

From the far-field condition we can determine

$$a = c = 0.$$

Furthermore, the velocity potential must be continuous at the symmetry plane and thus

$$b = d.$$

Kinematic condition

Finally, we transform the kinematic condition at the airfoil to obtain

$$\begin{cases} -\sqrt{M_\infty^2 - 1}\, f' = -2\xi & \text{at } -1 < \xi < 1 \;\text{ for } y > 0 \\ \sqrt{M_\infty^2 - 1}\, g' = \;\; 2\eta & \text{at } -1 < \eta < 1 \;\text{ for } y < 0 \end{cases},$$

which after integration leads to

$$\begin{cases} f = \left(M_\infty^2 - 1\right)^{-\frac{1}{2}} \xi^2 + j & \text{for } -1 < \xi < 1 \wedge y > 0 \\ g = \left(M_\infty^2 - 1\right)^{-\frac{1}{2}} \eta^2 + k & \text{for } -1 < \xi < 1 \wedge y < 0 \end{cases}.$$

Again, we must ensure a continuous velocity potential at the boundaries between different intervals by matching the integration constants. Along the characteristics $\xi = -1$ and $\eta = -1$ passing through leading edge we have, respectively,

$$f(-1) = \left(M_\infty^2 - 1\right)^{-\frac{1}{2}} (-1)^2 + j = a = 0,$$

$$g(-1) = \left(M_\infty^2 - 1\right)^{-\frac{1}{2}} (-1)^2 + k = c = 0,$$

so that

$$j = k = -\left(M_\infty^2 - 1\right)^{-\frac{1}{2}}.$$

Analogously, at $\xi = 1$ and $\eta = 1$ we match the constants

$$b = d = 0.$$

Result: Analytical velocity potential as a piecewise function

The analytical velocity potential reads

$$\tilde{\phi}_1 = \begin{cases} 0 & \text{for } \xi < -1 \cup \eta < -1 \cup (\xi > 1 \cap y > 0) \cup (\eta > 1 \cap y < 0) \\ \left(M_\infty^2 - 1\right)^{-\frac{1}{2}}(\xi^2 - 1) & \text{for } -1 < \xi < 1 \ \wedge y > 0 \\ \left(M_\infty^2 - 1\right)^{-\frac{1}{2}}(\eta^2 - 1) & \text{for } -1 < \eta < 1 \ \wedge \ y < 0 \end{cases} .$$

Defining the velocity potential in MATLAB as a symbolic piecewise function

```
% Intervals of the piecewise function
upstream   = xi <= -1 | eta <= -1 ;
downstream = (xi>=1 & y>=0) | (eta>=1 & y<=0) ;
upper      = xi >-1 & xi <1 & y>0 ;
lower      = eta>-1 & eta<1 & y<0 ;

% Solution in different sub-domains
phi_upper = 1./sqrt(M.^2-1) .* (xi .^2 -1) ;
phi_lower = 1./sqrt(M.^2-1) .* (eta.^2 -1) ;

% Global function
phi_1 = piecewise(upstream | downstream, 0, ...
                  upper,                 phi_upper, ...
                  lower,                 phi_lower)
```

phi_1 =

$$\begin{cases} 0 & \text{if } x + y\,\sigma_1 \le -1 \vee x + 1 \le y\,\sigma_1 \vee (1 \le x + y\,\sigma_1 \wedge y \le 0) \vee (y\,\sigma_1 + 1 \le x \wedge 0 \le y) \\ \dfrac{(x - y\,\sigma_1)^2 - 1}{\sigma_1} & \text{if } x < y\,\sigma_1 + 1 \wedge y\,\sigma_1 < x + 1 \wedge 0 < y \\ \dfrac{(x + y\,\sigma_1)^2 - 1}{\sigma_1} & \text{if } x + y\,\sigma_1 \in (-1, 1) \wedge y < 0 \end{cases}$$

where

$$\sigma_1 = \sqrt{M^2 - 1}$$

```
phi = phi_0 + epsilon*phi_1
```

phi =

$$\begin{cases} x & \text{if } x + y\,\sigma_1 \le -1 \vee x + 1 \le y\,\sigma_1 \vee (1 \le x + y\,\sigma_1 \wedge y \le 0) \vee (y\,\sigma_1 + 1 \le x \wedge 0 \le \\ x + \dfrac{\varepsilon\left((x - y\,\sigma_1)^2 - 1\right)}{\sigma_1} & \text{if } x < y\,\sigma_1 + 1 \wedge y\,\sigma_1 < x + 1 \wedge 0 < y \\ x + \dfrac{\varepsilon\left((x + y\,\sigma_1)^2 - 1\right)}{\sigma_1} & \text{if } x + y\,\sigma_1 \in (-1, 1) \wedge y < 0 \end{cases}$$

where

$$\sigma_1 = \sqrt{M^2 - 1}$$

Velocity components

```
u = diff(phi,x)
```

u =

$$\begin{cases} 1 & \text{if } x + y\,\sigma_1 < -1 \vee x + 1 < y\,\sigma_1 \vee (1 < x + y\,\sigma_1 \wedge y < 0) \vee (y\,\sigma_1 + 1 < x \wedge 0 < y) \\ \dfrac{\varepsilon\,(2\,x - 2\,y\,\sigma_1)}{\sigma_1} + 1 & \text{if } x < y\,\sigma_1 + 1 \wedge y\,\sigma_1 < x + 1 \wedge 0 < y \\ \dfrac{\varepsilon\,(2\,x + 2\,y\,\sigma_1)}{\sigma_1} + 1 & \text{if } x + y\,\sigma_1 \in (-1, 1) \wedge y < 0 \end{cases}$$

where

$$\sigma_1 = \sqrt{M^2 - 1}$$

```
v = diff(phi,y)
```

v =

$$\begin{cases} 0 & \text{if } x + \sigma_1 < -1 \vee x + 1 < \sigma_1 \vee (1 < x + \sigma_1 \wedge y < 0) \vee (\sigma_1 + 1 < x \wedge 0 < y) \\ -2\,\varepsilon\,(x - \sigma_1) & \text{if } x < \sigma_1 + 1 \wedge \sigma_1 < x + 1 \wedge 0 < y \\ 2\,\varepsilon\,(x + \sigma_1) & \text{if } x + \sigma_1 \in (-1, 1) \wedge y < 0 \end{cases}$$

where

$$\sigma_1 = y\,\sqrt{M^2 - 1}$$

Visualization

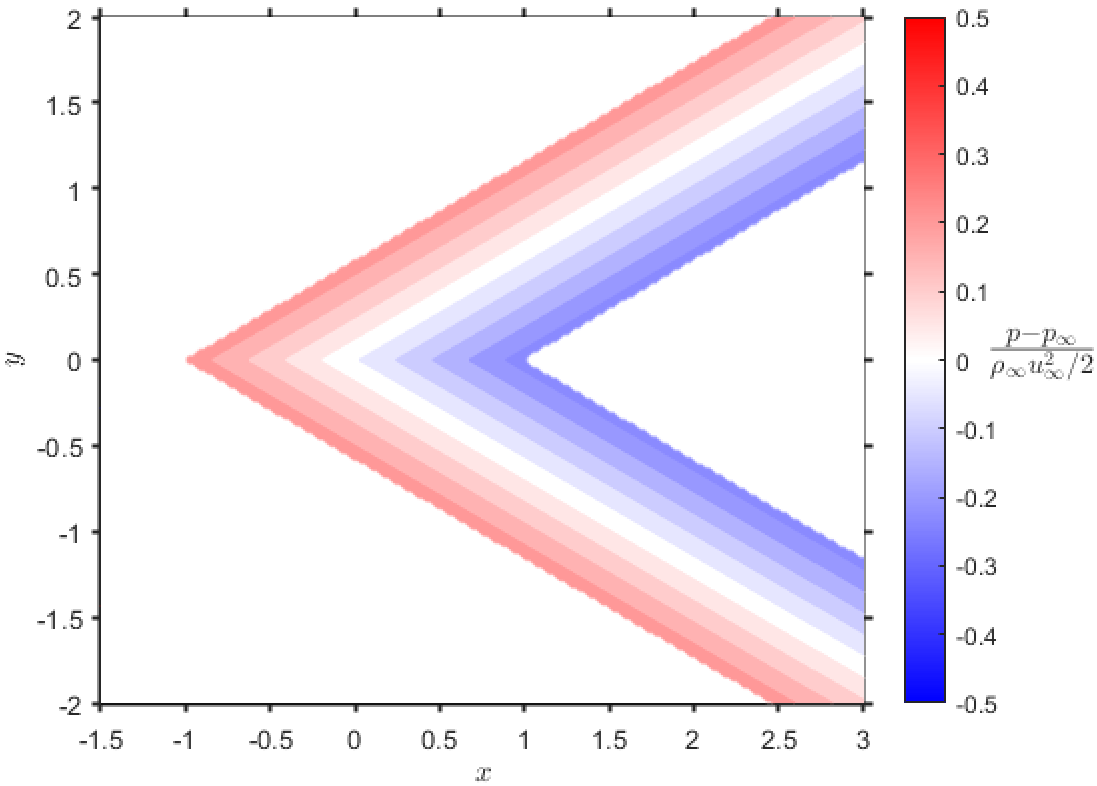
Let's try to plug in some values

```
% Parameters
M_val   = 2  ; % value of the Mach number
Epsilon = 0.1; % value of epsilon

% Range for plotting
xmin = -1.5;
xmax =  3  ;
ymin = -2  ;
ymax =  2  ;

% Resolution for velocity vectors
n = 10;

% Distribution of velocity vectors
[X,Y] = meshgrid(linspace(xmin,xmax,n),linspace(ymin,ymax,n));

% Substitute values into expressions
H   = Epsilon * h;
Phi = subs(phi,{M,epsilon},{M_val,Epsilon});
Xi  = subs(xi ,M,M_val);
Eta = subs(eta,M,M_val);
U   = subs(u,{x,y,M,epsilon},{X,Y,M_val,Epsilon});
```

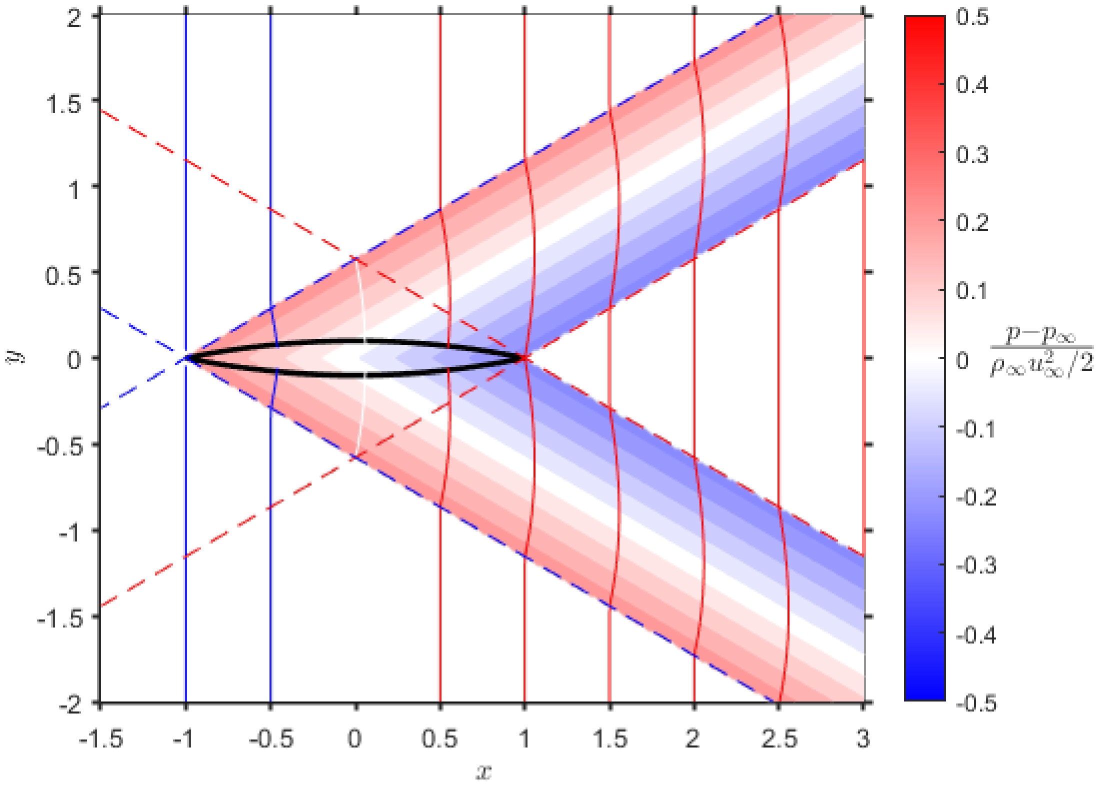
```
V   = subs(v,{x,y,M,epsilon},{X,Y,M_val,Epsilon});
P   = subs(2*(1-u), {M,epsilon}, {M_val,Epsilon});
```

and plot the solution.

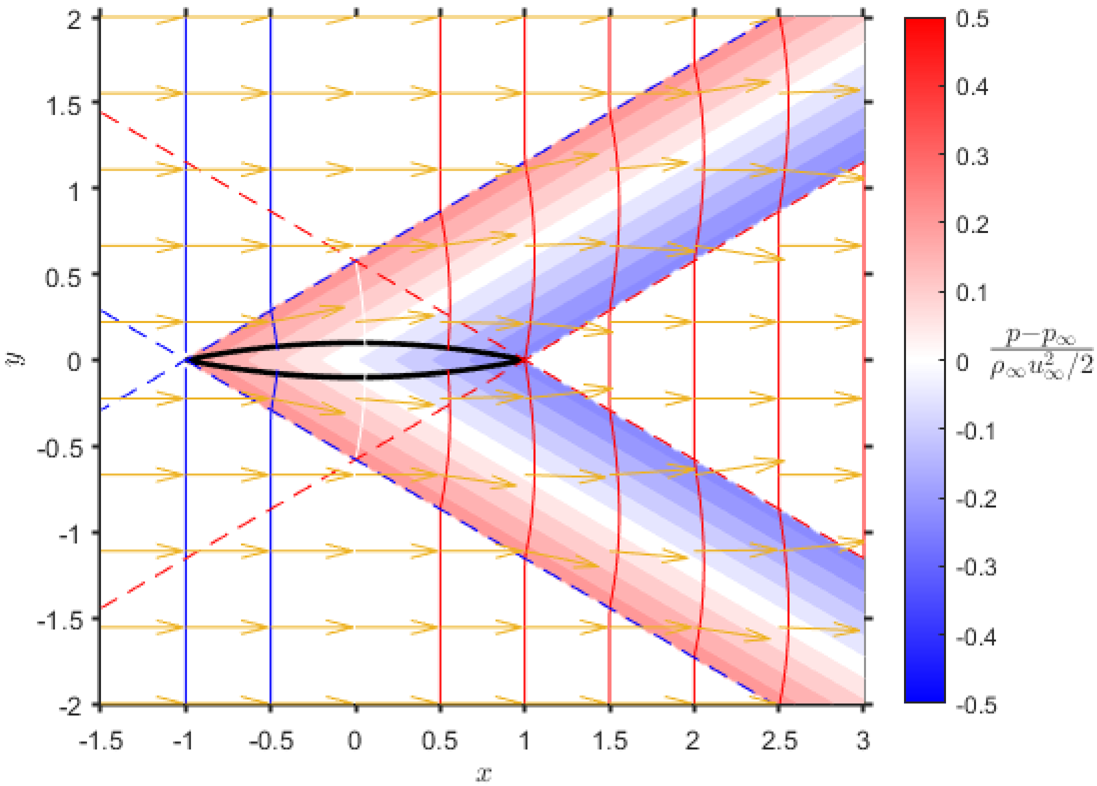
```
figure(1);
hold on;
box on;
set(gca,'TickDir','out','linewidth',1.5);
axis([xmin xmax ymin ymax]);
xlabel('$x$','Interpreter',"latex");
ylabel('$y$',"Interpreter","latex");
colormap(redblue);
clim([-0.5, 0.5]);

pressure = fcontour(P, [xmin xmax ymin ymax], 'Fill', 'on','MeshDensity',200);
cbar = colorbar;
cbar.Label.Interpreter = "latex";
cbar.Label.String = "$\frac{p-p_{\infty}}{\rho_{\infty} u_{\infty}^{2} / 2}$";
cbar.Label.Rotation = 0;
cbar.Label.FontSize = 15;
cbar.Label.Position = [3.3,0.05,0];
```




```
wing = fplot([H, -H], [-1 1], 'k-','LineWidth',2);
potential = fcontour(Phi, [xmin xmax ymin ymax]);
schock1 = fcontour(Xi , '--','LevelList',[-1 1]);
schock2 = fcontour(Eta, '--','LevelList',[-1 1]);
```




```
arrows = quiver(X,Y,U,V);
```



Analytical streamlines

Streamline $y_s(x)$ is a trajectory of an infinitely small fluid particle which moves with the steady flow. Therefore, the streamline is at all points tangential to the flow velocity. Streamlines can be computed, for example, by solving the initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}x} y_s(x) = \frac{v}{u}$$

for given initial conditions $y_s(x_0) = y_0$ corresponding to the initial locations of the fluid elements.

```
syms y_s(x)
f(x, y) = subs(v/u, {M,epsilon}, {M_val,Epsilon});
eq = diff(y_s,x) == f(x,y_s)
```

```
eq(x) =
```

$$\frac{\partial}{\partial x}\, y_s(x) = \begin{cases} 0 & \text{if } (0 < y_s(x) \wedge \sigma_1 + 1 < x) \vee (y_s(x) < 0 \wedge 1 < x + \sigma_1) \vee x + \sigma_1 < \\ -\dfrac{x - \sigma_1}{5\left(\dfrac{\sqrt{3}\,\,(2\,x - \sigma_2)}{30} + 1\right)} & \text{if } 0 < y_s(x) \wedge x < \sigma_1 + 1 \wedge \sigma_1 < x + 1 \\ \dfrac{x + \sigma_1}{5\left(\dfrac{\sqrt{3}\,\,(2\,x + \sigma_2)}{30} + 1\right)} & \text{if } y_s(x) < 0 \wedge x + \sigma_1 \in (-1, 1) \end{cases}$$

where

$$\sigma_1 = \sqrt{3}\,\, y_s(x)$$

$$\sigma_2 = 2\,\,\sqrt{3}\,\, y_s(x)$$

We may try to find an analytical solution for the streamlines as follows:

```
sol    = dsolve(eq)
```

```
Warning: Unable to find symbolic solution.
sol =
[ empty sym ]
```

```
sol = dsolve(eq,y_s(-1)==1)
```

```
Warning: Unable to find symbolic solution.
sol =
[ empty sym ]
```

In this case MATLAB didn't find any analytical solution, so we will instead solve the initial value problem numerically using the solver ode45.

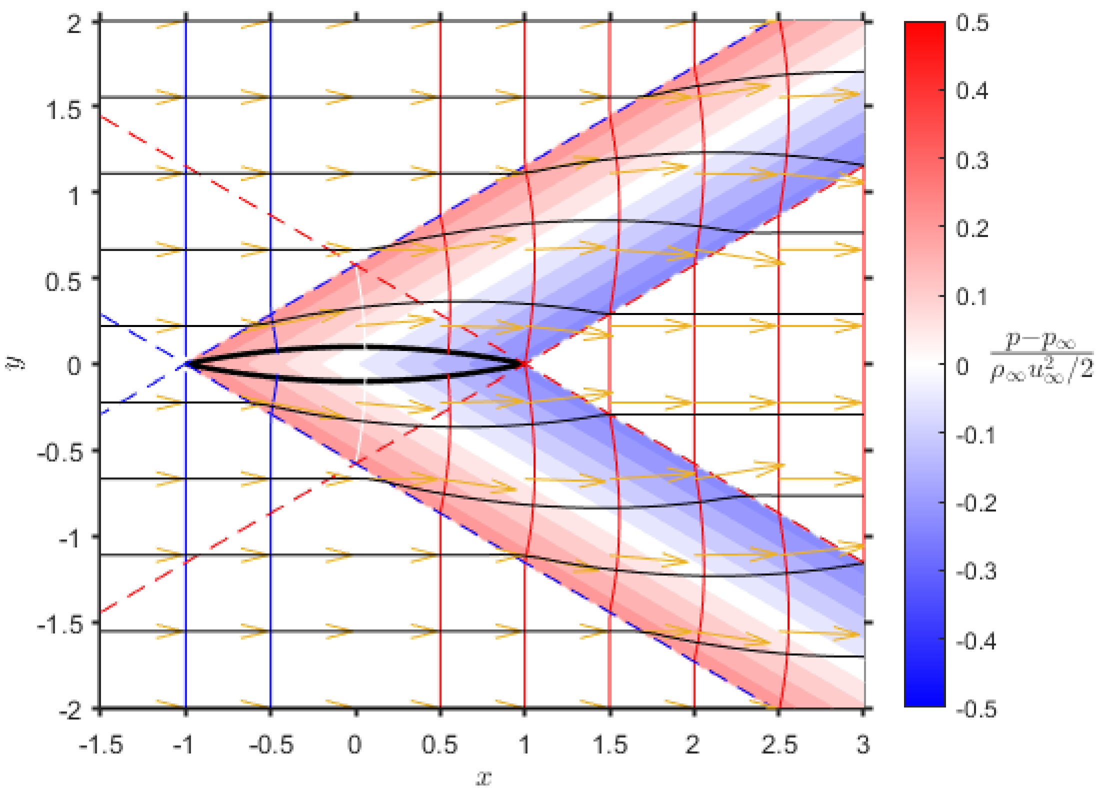
Numerical streamlines

First, we have to convert the right-hand side into a standard MATLAB function:

```
filename = 'streamline_slope.m';
F = odeFunction(f(x,y_s),y_s(x),'file',filename);
```

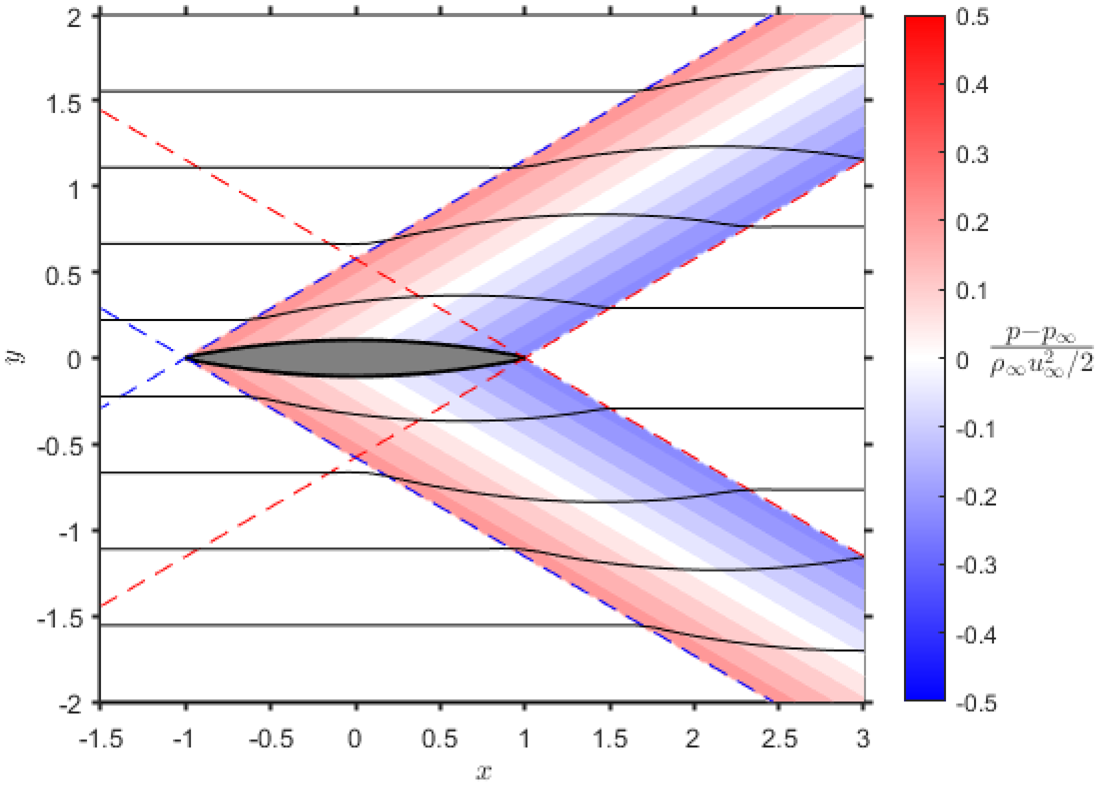
Then, we solve the initial value problem for a list of initial conditions and plot each resulting streamline.

```
for y0 = linspace(ymin,ymax,n)
    [X_s,Y_s] = ode45(F,[xmin,xmax],y0);
    plot(X_s,Y_s,'k-')
end
```



Finally, we clean up the plot and export it into a file.

```
potential.Visible = "off";
arrows.Visible = "off";
x_profile = -1:0.01:1;
y_profile = Epsilon*(-1+x_profile.^2);

x_profile = [x_profile, flip(x_profile)];
y_profile = [y_profile,-flip(y_profile)];

profile = fill(x_profile,y_profile,[0.5 0.5 0.5]);
```




```
saveas(gcf,'supersonic_flow.svg')
saveas(gcf,'supersonic_flow.png')
```