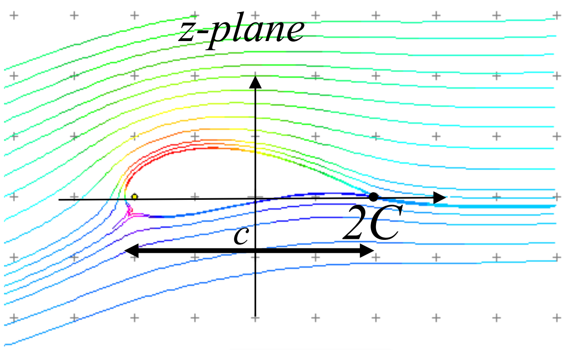
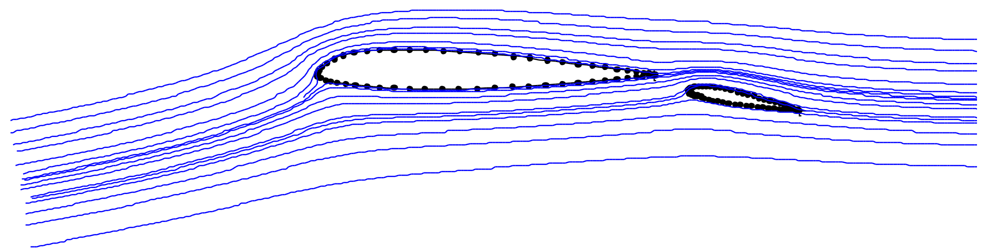
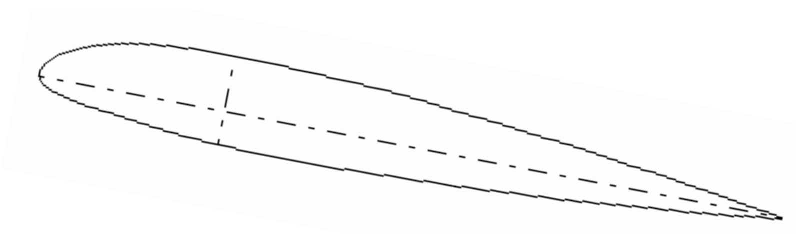
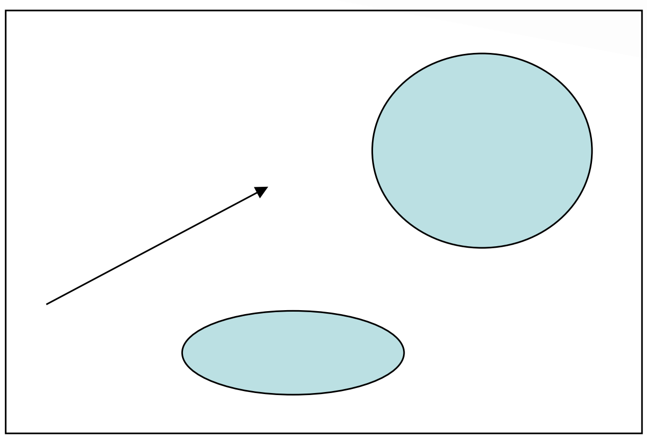
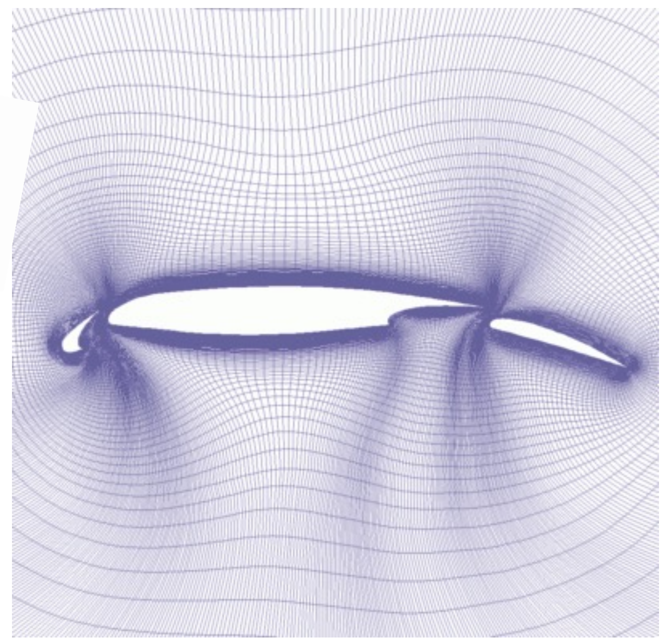
Panel Methods





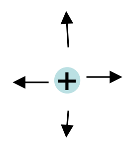






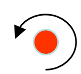
Source and Vortex

- Point Source



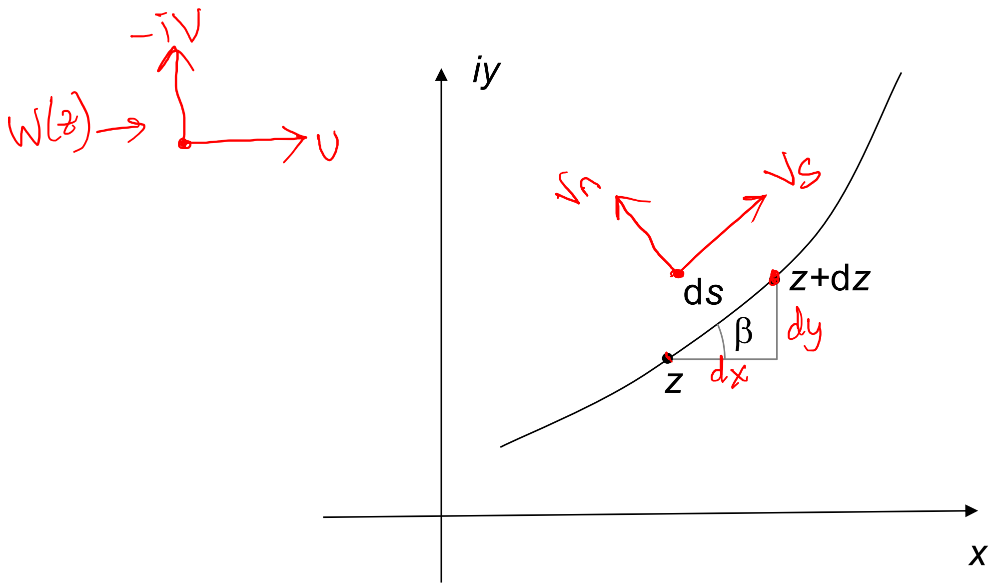
$$W(z) = \frac{q}{2\pi(z - z_1)}$$

- Point Vortex



$$W(z) = \frac{-i\Gamma}{2\pi(z - z_1)}$$

dz in Polar Form



$$dz = ds\, e^{i\beta} \quad \text{(1)}$$

$$\left[dx + i\,dy = ds\,.\,(\cos\beta + i\sin\beta) \right]$$

$$\Rightarrow \frac{dz}{ds} = e^{i\beta}$$

(or) $$\frac{ds}{dz} = e^{-i\beta}$$

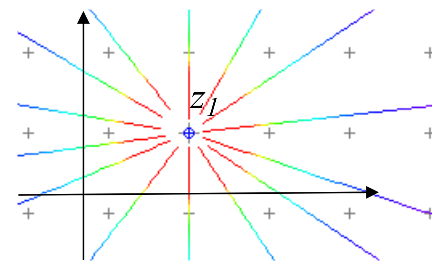
$$V_s - iV_n = W(z)\,.\,\frac{dz}{ds}$$

Panels

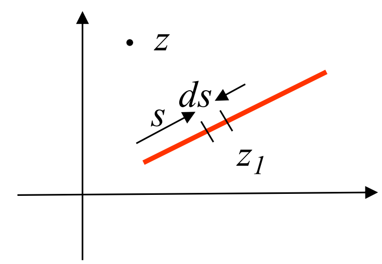
Singularity distributed along a line

Example: The Source Panel (or Sheet)

Consider a point source ⊕ $$W(z) = \frac{q}{2\pi(z - z_1)}$$

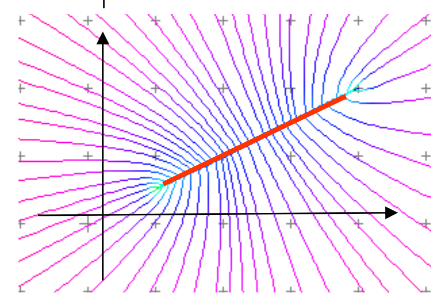


Imagine spreading the source along a line. We would then end up with a certain strength per unit length $q(s)$ that could vary with distance s along the line.

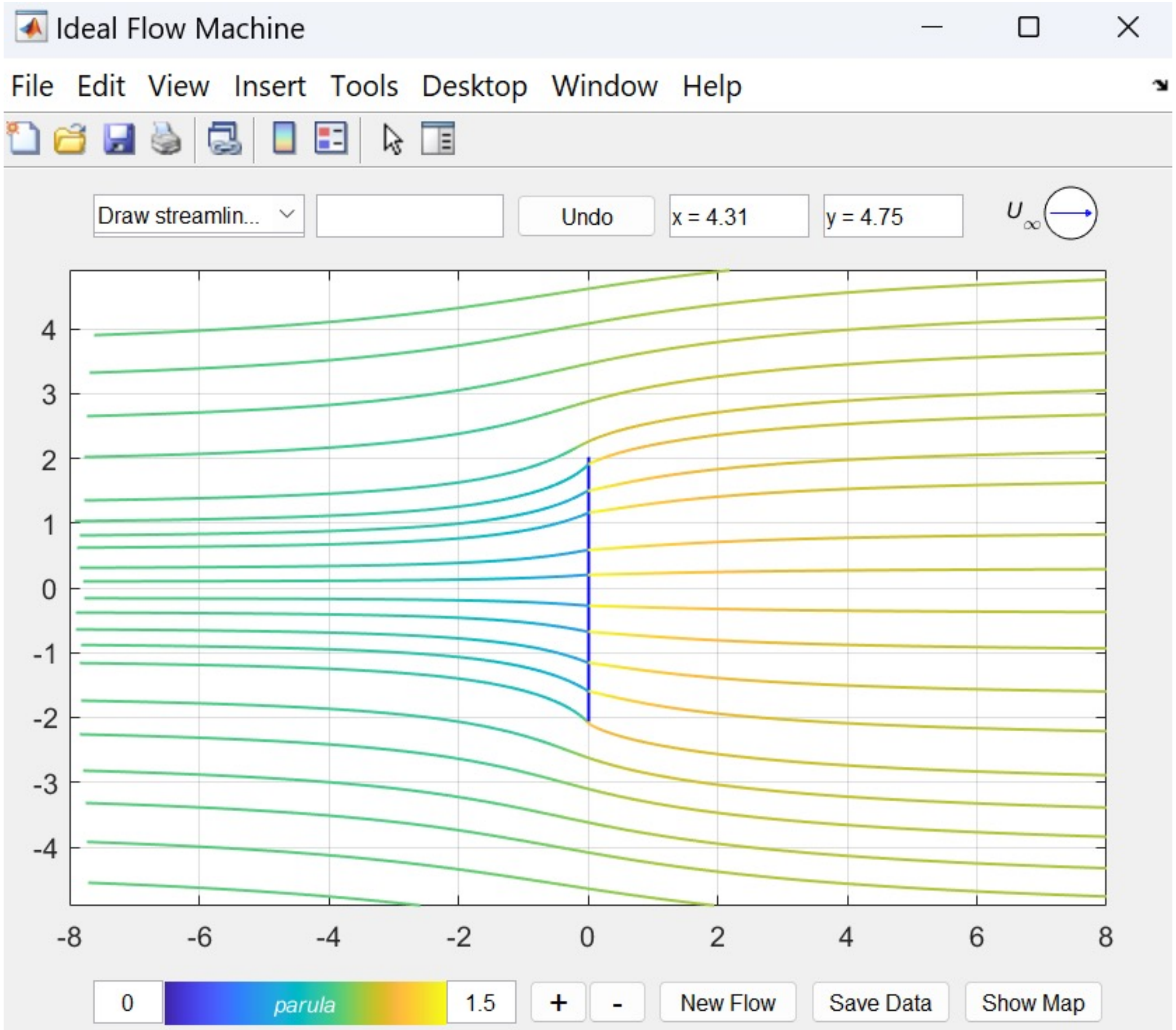


$$dW(z) = \frac{q\, ds}{2\pi(z - z_1)}$$

FOR WHOLE PANEL:



$$W(z) = \frac{1}{2\pi}\int_{\text{PANEL}} \frac{q(s).ds}{z - z_1(s)}$$

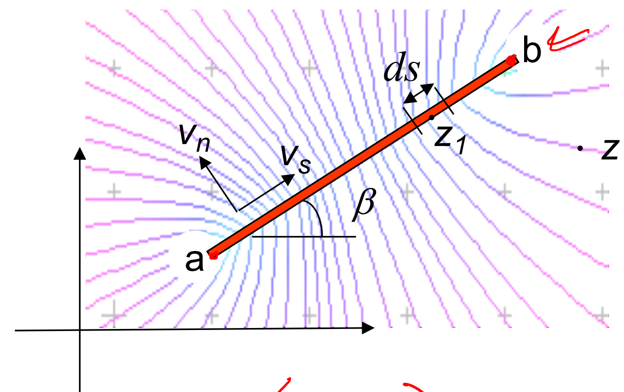

Ideal Flow Machine
File Edit View Insert Tools Desktop Window Help
Draw streamlin...
Undo
x = 4.31
y = 4.75
U_∞
4
3
2
1
0
-1
-2
-3
-4
-8 -6 -4 -2 0 2 4 6 8
0
parula
1.5
+
-
New Flow
Save Data
Show Map

Constant Strength Source Panel

$$W(z) = \frac{1}{2\pi}\int_{\text{PANEL}} \frac{q(s)\,ds}{z - z_1(s)}$$

$q(s) = q = $ CONSTANT

PANEL IS STRAIGHT



$$W(z) = \frac{q}{2\pi}\int_{\text{PANEL}} \frac{ds}{z - z_1(s)} = \frac{q}{2\pi}\int_{\text{PANEL}} \frac{dz_1}{z - z_1(s)} \cdot \underbrace{\frac{ds}{dz_1}}_{= e^{-i\beta}}$$

$$W(z) = -\frac{q}{2\pi}\log_e(z - z_1)\Big|_a^b \frac{ds}{dz_1} = \frac{q}{2\pi}\log_e(z - z_1)\Big|_b^a \frac{ds}{dz_1} = \frac{q}{2\pi}\log_e\left(\frac{z - z_a}{z - z_b}\right)\cdot\frac{ds}{dz_1}$$

(OR)

$$V_s - iV_n = W(z)\cdot\frac{dz_1}{ds} = \frac{q}{2\pi}\log_e\left(\frac{z - z_a}{z - z_b}\right)$$

$\frac{q}{2\pi} \rightarrow -i\Gamma$ for Vortex panel.

The panel is not a solid boundary to the flow. To make it behave like one you would set the strength q so that the total v_n (due to the panel and the flow) is zero

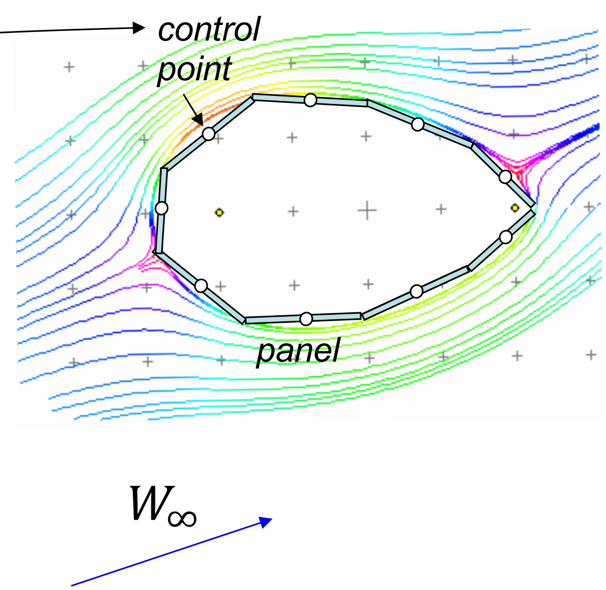
Vortex?

A Simple Source Panel Method

For flow past an arbitrary body

N UNKNOWNS ($q_i \rightarrow$ 1 to N)

- Break up the body surface into *N* straight panels.
- Write an expression for the normal component of velocity at the middle of the panel from the sum of all the velocities produced by the panels and the free stream. Gives *N* expressions.
- Given that each expression must be equal to zero, solve the *N* equations for the *N* strengths



Defining the N Panels

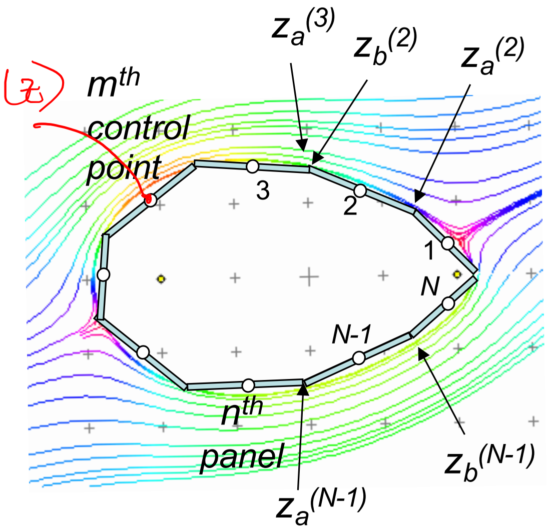
- Number the panels anticlockwise, 1 to N
- Define N coordinates z_a that identify the start of every panel (going counter clockwise), and z_b that identify the end of every panel.

- Each panel has a slope $\dfrac{dz}{ds} = e^{i\beta} = \dfrac{z_b - z_a}{|z_b - z_a|}$

So, if $W(z)$ is the velocity of the whole flow,

$-\operatorname{Im}\left\{W(z)\dfrac{dz}{ds}\right\}$ is the component normal to the panel

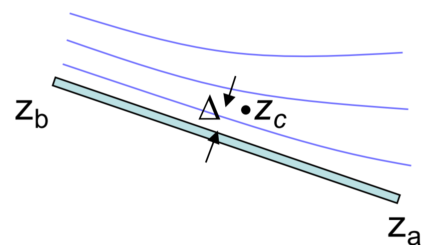
W(z)



- We pick a control point very close to the center of the panel at

$$z_c = \frac{1}{2}(z_a + z_b) - i\varepsilon(z_b - z_a)$$

Center point *Displacement Δ*



ε<<1 (say 0.001)

Completing the Method

Velocity produced by whole flow is

$$W(z) = W_\infty + \sum_{n=1}^{N} q^{(n)} \frac{1}{2\pi} \log_e \left(\frac{z - z_a{}^{(n)}}{z - z_b{}^{(n)}} \right) \frac{ds}{dz_1}\bigg|^{(n)} \tag{1}$$

Velocity at the control point of the m^th panel $z_c{}^{(m)}$ in panel aligned components is

$$= W_\infty \frac{dz_1}{ds}\bigg|^{(m)} + \sum_{n=1}^{N} q^{(n)} \underbrace{\frac{1}{2\pi} \log_e \left(\frac{z_c{}^{(m)} - z_a{}^{(n)}}{z_c{}^{(m)} - z_b{}^{(n)}} \right) \frac{ds}{dz_1}\bigg|^{(n)} \frac{dz_1}{ds}\bigg|^{(m)}}_{C^{(m,n)}}$$

So, velocity normal to the m^th panel is

$$= \mathrm{Im}\left\{ \frac{dz_1}{ds}\bigg|^{(m)} W_\infty \right\} + \sum_{n=1}^{N} q^{(n)}\, \mathrm{Im}\{ C^{(m,n)} \}$$

Velocity parallel to the m^th panel is

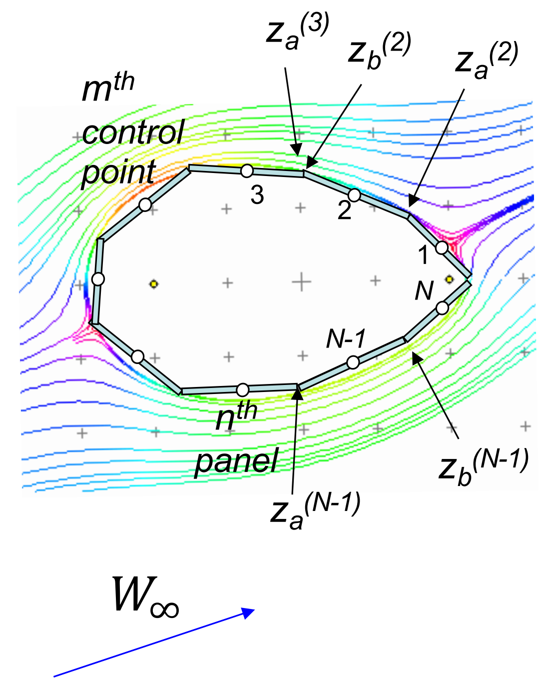
$$= \mathrm{Re}\left\{ \frac{dz_1}{ds}\bigg|^{(m)} W_\infty \right\} + \sum_{n=1}^{N} q^{(n)}\, \mathrm{Re}\{ C^{(m,n)} \} \tag{2}$$

We want the normal velocity to be zero, so this is what we use to get the *q*'s

We write $$-\mathrm{Im}\left\{ \frac{dz_1}{ds}\bigg|^{(m)} W_\infty \right\} = \sum_{n=1}^{N} q^{(n)}\, \mathrm{Im}\{ C^{(m,n)} \}$$

Or 1xN result matrix (known) = 1xN matrix of strengths (unknown) x NxN matrix of ceoffs (known)

Once we have solved this for the *q*'s we can use eqn. 2 to get the velocities along the body surface, or eqn. 1 to get them anywhere else

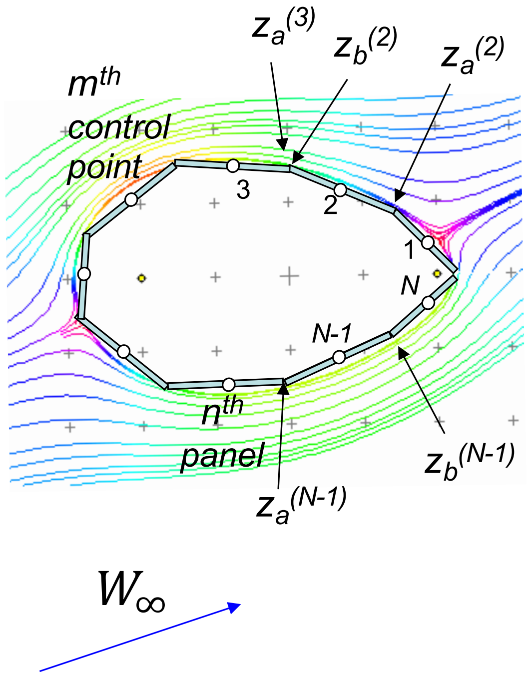


Computational Steps

- Define coordinates of start and end of panels z_a and z_b $\boxed{z_a^{(n)}, z_b^{(n)}}$
- Compute the panel slopes $\boxed{\frac{dz}{ds} = e^{i\beta} = \frac{z_b - z_a}{|z_b - z_a|}}$
- Put the control points next to the panel centers $\boxed{z_c = 1/2(z_a + z_b) - i\varepsilon(z_b - z_a)}$
- Determine the influence coefficients

$$\boxed{C^{(m,n)} = \frac{1}{2\pi}\log_e\left(\frac{z_c^{(m)} - z_a{}^{(n)}}{z_c^{(m)} - z_b{}^{(n)}}\right)\frac{ds}{dz_1}\Bigg|^{(n)}\frac{dz_1}{ds}\Bigg|^{(m)}}$$

- Determine the component of W_∞ normal to each panel $\boxed{-\operatorname{Im}\left\{\frac{dz_1}{ds}\Bigg|^{(m)} W_\infty\right\}}$
- Solve the matrix problem, i.e. matrix divide . $\boxed{\operatorname{Im}\{C^{(m,n)}\}}$ by $\boxed{-\operatorname{Im}\left\{\frac{dz_1}{ds}\Bigg|^{(m)} W_\infty\right\}}$
- Compute the flow velocities and pressures



Computational Steps

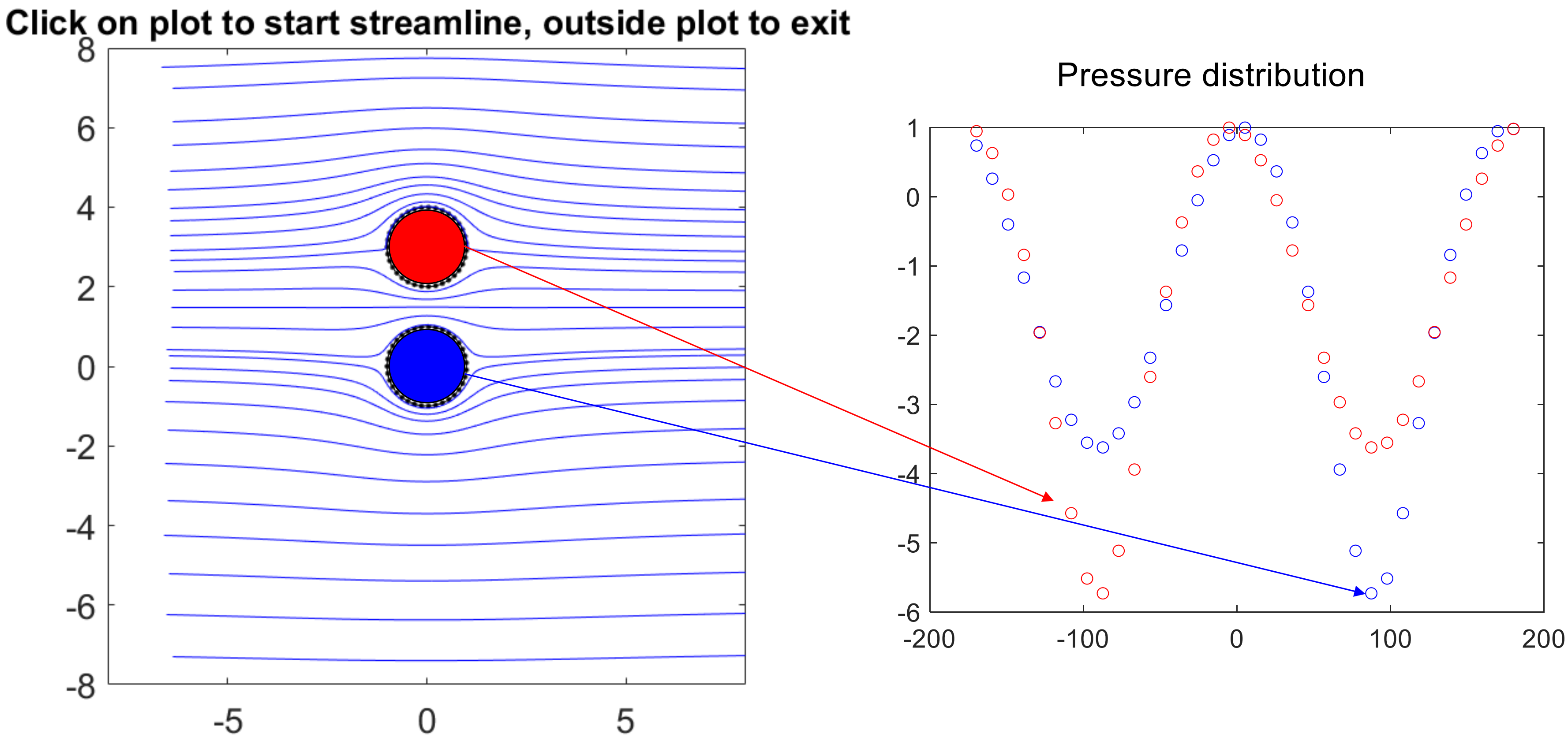
ConstantSourcePanel.m

- Define coordinates of start and end of panels z_a and z_b $\quad z_a^{(n)}, z_b^{(n)}$
- Compute the panel slopes $\quad \dfrac{dz}{ds} = e^{i\beta} = \dfrac{z_b - z_a}{|z_b - z_a|}$
- Put the control points next to the panel centers $\quad z_c = 1/2(z_a + z_b) - i\varepsilon(z_b - z_a)$
- Determine the influence coefficients

$$C^{(m,n)} = \frac{1}{2\pi}\log_e\left(\frac{z_c^{(m)} - z_a^{(n)}}{z_c^{(m)} - z_b^{(n)}}\right)\frac{ds}{dz_1}\bigg|^{(n)} \frac{dz_1}{ds}\bigg|^{(m)}$$

- Determine the component of W_∞ normal to each panel $\quad -\operatorname{Im}\left\{\dfrac{dz_1}{ds}\bigg|^{(m)} W_\infty\right\}$
- Solve the matrix problem, i.e. matrix divide . $\operatorname{Im}\{C^{(m,n)}\}$ by $-\operatorname{Im}\left\{\dfrac{dz_1}{ds}\bigg|^{(m)} W_\infty\right\}$
- Compute the flow velocities and pressures

```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
    cm(:,m)=log((zc(m)-z(a))./(zc(m)-z(b)))/2/pi./dzds(a)
end
res=imag(-winf*dzds);
q=res/imag(cm);

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
plot(angle(zc)*180/pi,cp);
```


Click on plot to start streamline, outside plot to exit
8
6
4
2
0
-2
-4
-6
-8
-5
0
5
Pressure distribution
1
0
-1
-2
-3
-4
-5
-6
-200
-100
0
100
200

ConstantSourcePanel.m

Matlab Code

$z_a^{(n)}, z_b^{(n)}, dz_1/ds\big|^{(n)}$

$z_c^{(n)}$

$C^{(m,n)}$

Matrix div.

```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
    cm(:,m)=log((zc(m)-z(a))./(zc(m)-z(b)))/2/pi./dzds(a)*dzds(m);
end
res=imag(-winf*dzds);
q=res/imag(cm);

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
plot(angle(zc)*180/pi,cp);
```

$-\operatorname{Im}\left\{ \frac{dz_1}{ds}\bigg|^{(m)} W_\infty \right\}$

Result matrix

$\operatorname{Re}\left\{ \frac{dz_1}{ds}\bigg|^{(m)} W_\infty \right\} + \sum_{n=1}^{N} q^{(n)} \operatorname{Re}\{C^{(m,n)}\}$

Velocities along body surface

ConstantSourcePanel.m

Matlab Code Ideas:
1. Non-Uniform Free Stream

E.g. Suppose free stream includes, say a doublet outside the body at a location x=5, so

$$W_\infty = 1 + \frac{10}{(z-5)^2}$$

```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
    cm(:,m)=log((zc(m)-z(a))./(zc(m)-z(b)))/2/pi./dzds(a)*dzds(m);
end
res=imag(-winf*dzds);
q=res/imag(cm);

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
plot(angle(zc)*180/pi,cp);
```

ConstantSourcePanel.m

Matlab Code Ideas:
3. Use more sophisticated panels

E.g. Panels with linearly varying strength

```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
    cm(:,m)=log((zc(m)-z(a))./(zc(m)-z(b)))/2/pi./dzds(a)*dzds(m);
end
res=imag(-winf*dzds);
q=res/imag(cm);

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
plot(angle(zc)*180/pi,cp);
```

LinearSourcePanel.m

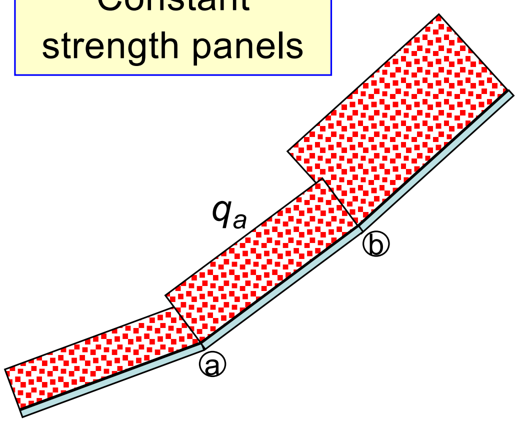
3. Linear Source Panel Method

E.g. Panels with linearly varying strength

```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];c=[3:npanels 1 2];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
cm(:,m)=(((zc(m)-z(a))./(z(b)-z(a)).*log((zc(m)-z(a))./(zc(m)-z(b)))
        -((zc(m)-z(c))./(z(b)-z(c)).*log((zc(m)-z(c))./(zc(m)-z(b)))
end
res=imag(-winf*dzds);
q=res/imag(cm);

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
plot(angle(zc)*180/pi,cp);
```

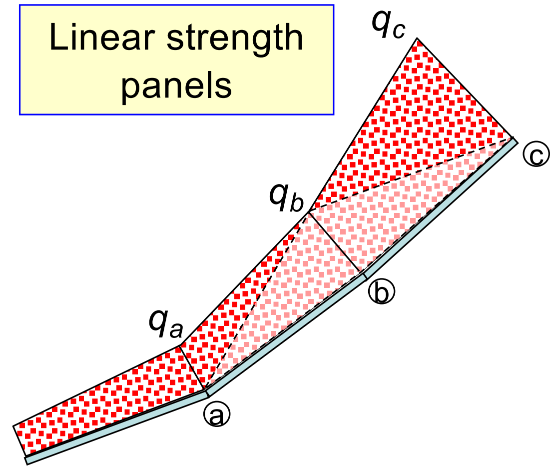
3. Linear Source Panels

Constant strength panels



Influence of q_a depends only panel ab

Linear strength panels



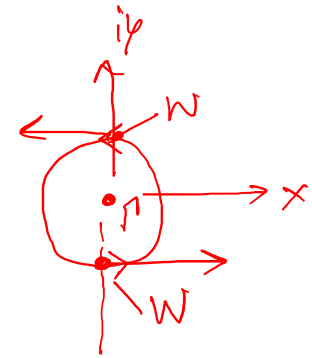
Influence of q_b depends on panel ab and panel bc

LinearSourcePanel.m

3. Linear Source Panel Method

E.g. Panels with linearly varying strength

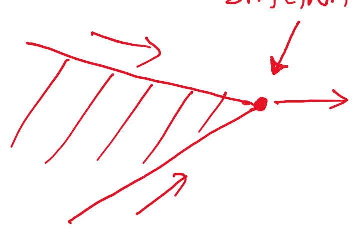
```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];c=[3:npanels 1 2];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
cm(:,m)=(((zc(m)-z(a))./(z(b)-z(a)).*log((zc(m)-z(a))./(zc(m)-z(b)))
        -((zc(m)-z(c))./(z(b)-z(c)).*log((zc(m)-z(c))./(zc(m)-z(b)))
end
res=imag(-winf*dzds);
q=res/imag(cm);

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
plot(angle(zc)*180/pi,cp);
```



30P30N 3-element high lift subsonic airfoil

Kutta condition

CIRCULATION AT T.E = 0



WEDGE T.E

FLOW VELOCITY AND PRESSURE MUST BE SAME ON BOTH SIDES OF T.E

CUSPED T.E

LinearSourcePanel.m

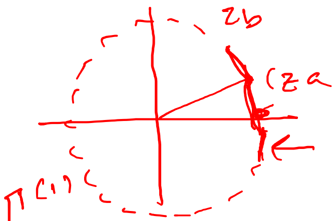
Matlab Code Ideas:
4.Change to vortex panel method

$$W(z) = \frac{q}{2\pi(z - z_1)}$$

$$W(z) = \frac{-i\Gamma}{2\pi(z - z_1)}$$

```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];c=[3:npanels 1 2];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
cm(:,m)=(((zc(m)-z(a))./(z(b)-z(a)).*log((zc(m)-z(a))./(zc(m)-z(b)))
        -((zc(m)-z(c))./(z(b)-z(c)).*log((zc(m)-z(c))./(zc(m)-z(b)))
end
res=imag(-winf*dzds);
q=res/imag(cm);

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
plot(angle(zc)*180/pi,cp);
```

$$-\,\mathrm{Im}\left\{\left.\frac{dz_1}{ds}\right|^{(m)} W_\infty\right\} = \sum_{n=1}^{N} q^{(n)}\,\mathrm{Im}\{C^{(m,n)}\}$$

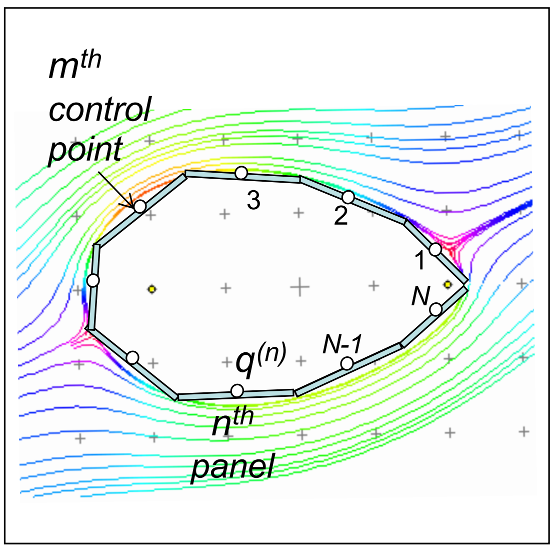


$$\begin{pmatrix} res(1) \\ res(2) \\ \vdots \\ res(m) \\ \vdots \\ \cancel{res(N)} \end{pmatrix} = \begin{pmatrix} \mathrm{Im}\{C^{(1,1)}\} & \mathrm{Im}\{C^{(1,2)}\} & \cdots & \mathrm{Im}\{C^{(1,n)}\} & \cdots & \mathrm{Im}\{C^{(1,N)}\} \\ \mathrm{Im}\{C^{(2,1)}\} & \cdot & & & & \\ \vdots & & \cdot & & & \\ \mathrm{Im}\{C^{(m,1)}\} & & & \mathrm{Im}\{C^{(m,n)}\} & & \\ \vdots & & & & \cdot & \\ \mathrm{Im}\{C^{(N,1)}\} & & & & & \mathrm{Im}\{C^{(N,N)}\} \end{pmatrix} \begin{pmatrix} q^{(1)} \\ q^{(2)} \\ \vdots \\ q^{(n)} \\ \vdots \\ q^{(N)} \end{pmatrix}$$

$0 \quad = \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0$

$\Gamma(1)$
$\Gamma(2)$
$\Gamma(3)$
$\vdots$
$\Gamma(n)$

$0 = \Gamma_{\text{end panel}}$



LinearVortexPanelKutta.m

Matlab Code Ideas:
5. Kutta Condition Code

Kutta condition requires that surface vorticity at trailing edge is zero.

```
clear all;
%Circular cylinder example, radius 2, 35 panels
npanels=35;r=2;winf=1;k=npanels-1;
z=r*exp(i*[2*pi/npanels:2*pi/npanels:2*pi]);
a=[1:npanels];b=[2:npanels 1];c=[3:npanels 1 2];
dzds=(z(b)-z(a))./abs(z(b)-z(a));

eps=0.0001;
zc=(z(a)+z(b))/2-i*eps*(z(b)-z(a)); %control points

cm=zeros(npanels);
for m=1:npanels
cm(:,m)=-i*(((zc(m)-z(a))./(z(b)-z(a)).*log((zc(m)-z(a))./(zc(m)-z(b
          -((zc(m)-z(c))./(z(b)-z(c)).*log((zc(m)-z(c))./(zc(m)-z(b
end
res=imag(-winf*dzds);
cm1=imag(cm);cm1(:,end)=0;cm1(k,end)=1;
res(end)=0;
q=res/cm1;

ut=real(q*cm+winf*dzds);
cp=1-ut.^2/abs(winf).^2;
figure
```

n-1

Things to watch out for…

- Always use an odd number of panels when you have a symmetric body (??)
- The control-point equation

$$z_c = \underbrace{\frac{1}{2}(z_a + z_b)}_{\textit{Center point}} - \underbrace{i\varepsilon(z_b - z_a)}_{\textit{Displacement } \Delta}$$

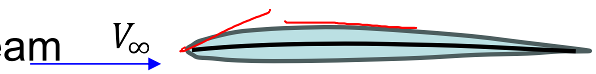
assumes that as you move from a to b you are progressing counter-clockwise around the body surface. (For clockwise you need to reverse the sign before i).

Thin Airfoil Theory

Beech Musketeer with tip removed showing airfoil

Low-speed, **thin** airfoils

Consider a **thin** arbitrary shape in freestream



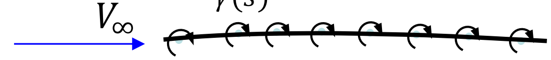
Assumptions:

1) Airfoil is thin
   (thickness << chord)

2) Angle, slopes are small
   (slopes ~ angles)
   $\tan\theta \sim \theta$

3) Airfoil only slightly disturbs the free-stream
   (u', v' << V_∞)

1. Approximate the airfoil shape with the airfoil mean camber line and place a vortex sheet $\gamma(s)$



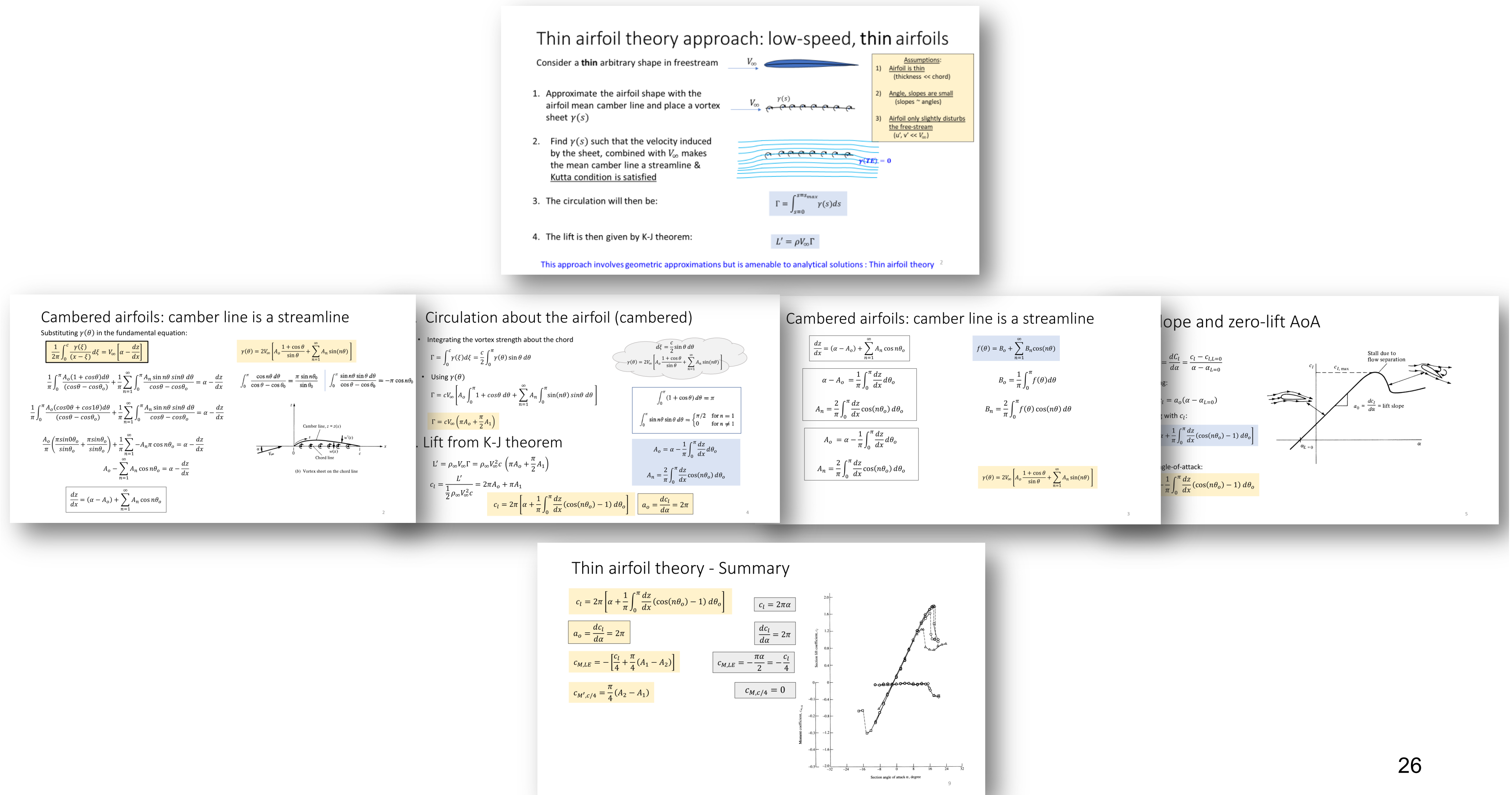
2. Find $\gamma(s)$ such that the velocity induced by the sheet, combined with V_∞ makes the mean camber line a streamline & Kutta condition is satisfied

3. The circulation will then be:

$$\Gamma = \int_{s=0}^{s=s_{max}} \gamma(s)ds$$

4. The lift is then given by K-J theorem:

$$L' = \rho V_\infty \Gamma$$



Thin airfoil theory approach: low-speed, thin airfoils
Consider a thin arbitrary shape in freestream
1. Approximate the airfoil shape with the airfoil mean camber line and place a vortex sheet $\gamma(s)$
2. Find $\gamma(s)$ such that the velocity induced by the sheet, combined with V_∞ makes the mean camber line a streamline & Kutta condition is satisfied
3. The circulation will then be:
$\Gamma = \int_{s=0}^{s=s_{max}} \gamma(s) ds$
4. The lift is then given by K-J theorem:
$L' = \rho V_\infty \Gamma$
Assumptions:
1) Airfoil is thin (thickness << chord)
2) Angle, slopes are small (slopes ~ angles)
3) Airfoil only slightly disturbs the free-stream (u', v' << V_∞)
$\gamma(TE) = 0$
This approach involves geometric approximations but is amenable to analytical solutions : Thin airfoil theory
Cambered airfoils: camber line is a streamline
Substituting $\gamma(\theta)$ in the fundamental equation:
Camber line, z = z(x)
Chord line
(b) Vortex sheet on the chord line
Circulation about the airfoil (cambered)
Integrating the vortex strength about the chord
Using $\gamma(\theta)$
$\Gamma = cV_\infty\left(\pi A_o + \frac{\pi}{2}A_1\right)$
Lift from K-J theorem
$c_l = 2\pi A_o + \pi A_1$
$c_l = 2\pi\left[\alpha + \frac{1}{\pi}\int_0^\pi \frac{dz}{dx}(\cos(n\theta_o) - 1)\, d\theta_o\right]$
$a_o = \frac{dc_l}{d\alpha} = 2\pi$
Cambered airfoils: camber line is a streamline
$\frac{dz}{dx} = (\alpha - A_o) + \sum_{n=1}^{\infty} A_n \cos n\theta_o$
$A_o = \alpha - \frac{1}{\pi}\int_0^\pi \frac{dz}{dx} d\theta_o$
$A_n = \frac{2}{\pi}\int_0^\pi \frac{dz}{dx}\cos(n\theta_o)\, d\theta_o$
$f(\theta) = B_o + \sum_{n=1}^{\infty} B_n \cos(n\theta)$
lope and zero-lift AoA
Stall due to flow separation
$a_o = \frac{dc_l}{d\alpha}$ = lift slope
Thin airfoil theory - Summary
$c_l = 2\pi\left[\alpha + \frac{1}{\pi}\int_0^\pi \frac{dz}{dx}(\cos(n\theta_o) - 1)\, d\theta_o\right]$
$c_l = 2\pi\alpha$
$a_o = \frac{dc_l}{d\alpha} = 2\pi$
$\frac{dc_l}{d\alpha} = 2\pi$
$c_{M,LE} = -\left[\frac{c_l}{4} + \frac{\pi}{4}(A_1 - A_2)\right]$
$c_{M,LE} = -\frac{\pi\alpha}{2} = -\frac{c_l}{4}$
$c_{M',c/4} = \frac{\pi}{4}(A_2 - A_1)$
$c_{M,c/4} = 0$

Thin airfoil theory - Results

Angle of attack

$\alpha - \alpha_o$

$(\alpha_o = 0$ if symmetric$)$

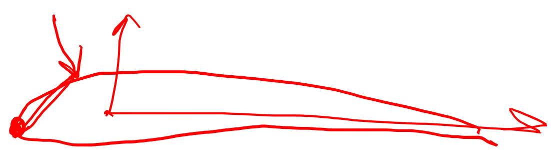
$$c_l = 2\pi\left[\alpha + \overbrace{\frac{1}{\pi}\int_0^{\pi}\frac{dz}{dx}(\cos(n\theta_o) - 1)\,d\theta_o}^{-\alpha_{L=0} \text{ or } \alpha_o}\right]$$

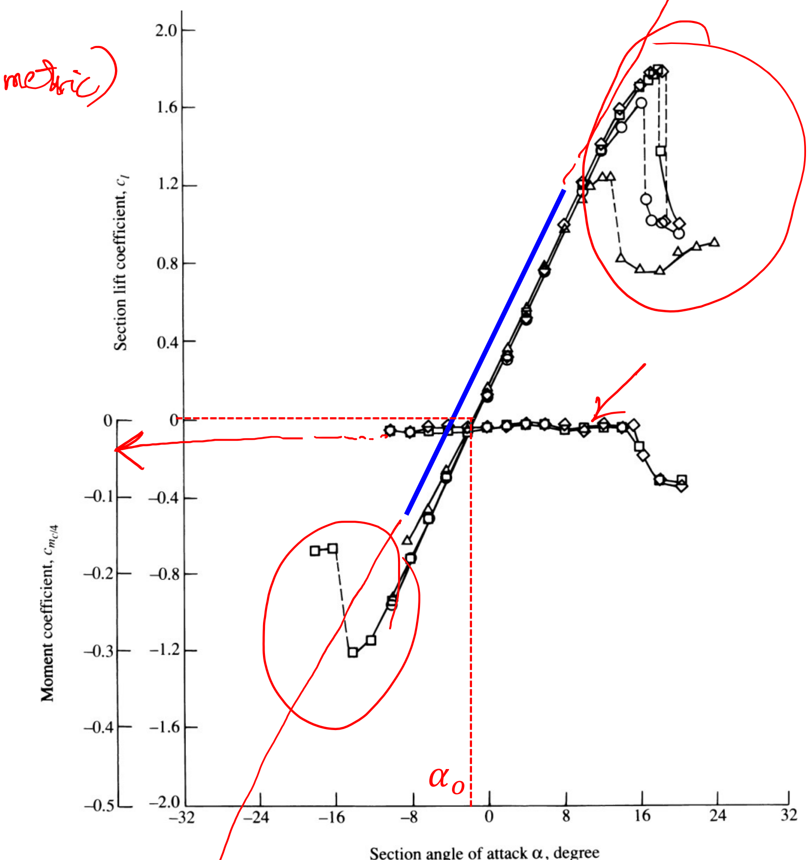
$$\boxed{\frac{dc_l}{d\alpha} = 2\pi}$$

f(camber)

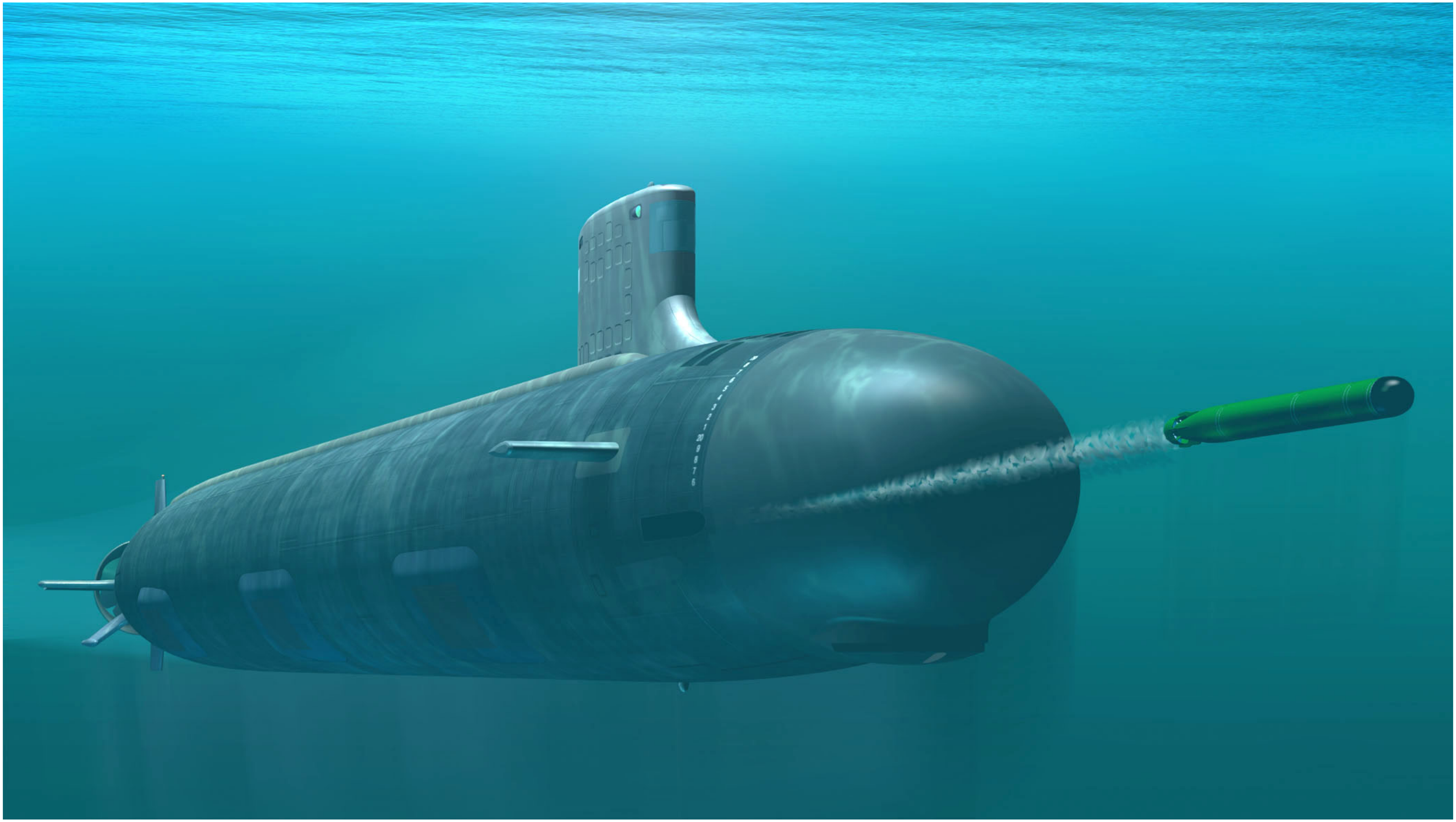
$$c_{m,LE} = -\left[\frac{c_l}{4} + \frac{\pi}{4}(A_1 - A_2)\right]$$

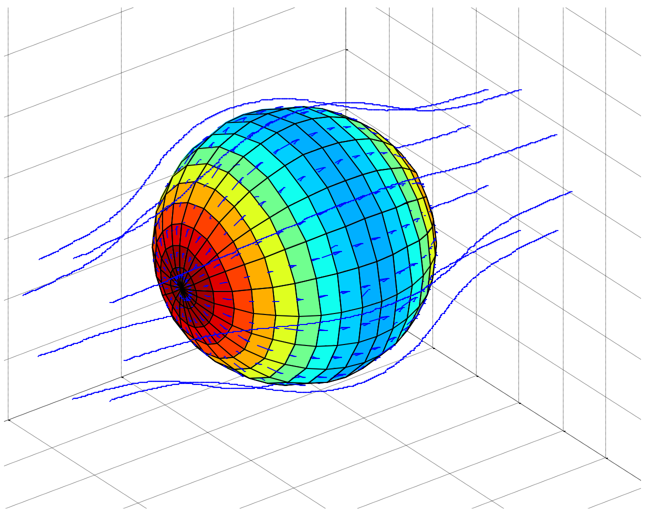
$$c_{m',c/4} = \frac{\pi}{4}(A_2 - A_1)$$

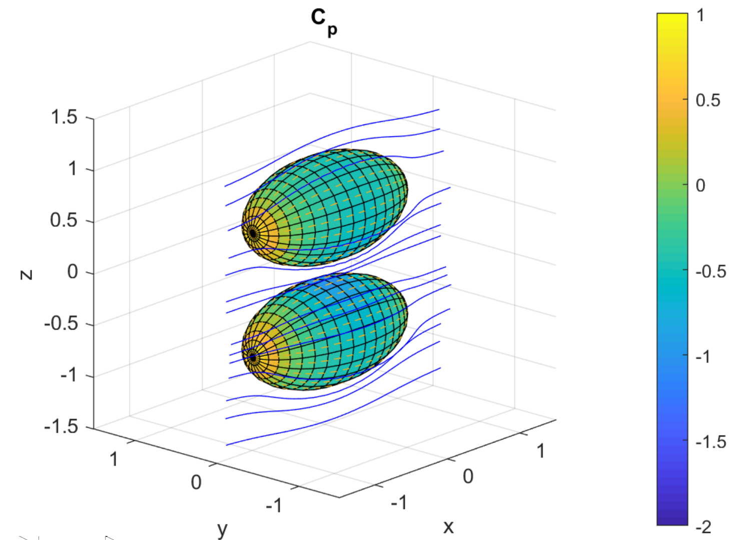


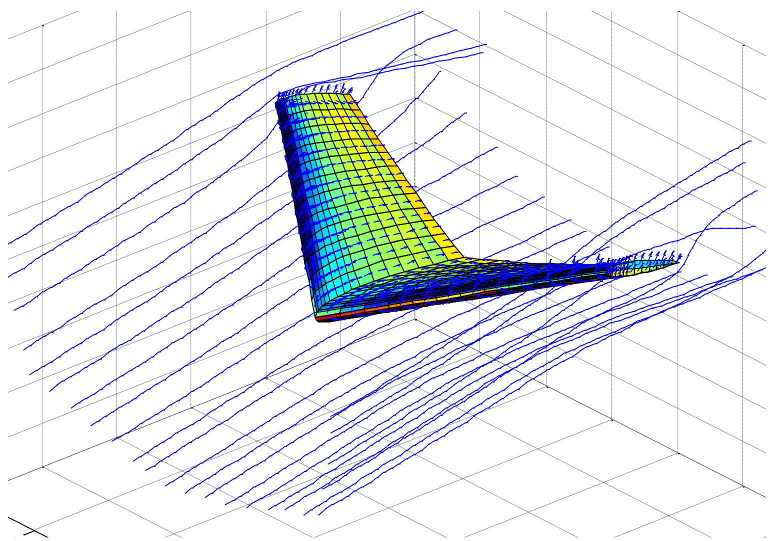



859
NASA
X-59
X-59






C_p
1.5
1
0.5
0
-0.5
-1
-1.5
z
1
0
-1
y
-1
0
1
x
1
0.5
0
-0.5
-1
-1.5
-2

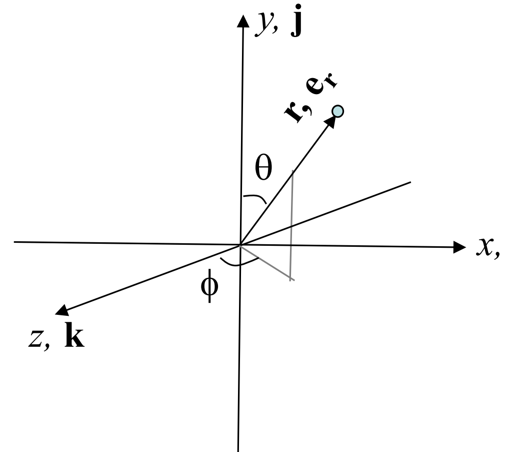


3D Steady Ideal Flow

IRROTATIONALITY : $\vec{\nabla} \times \vec{V} = 0 \Rightarrow \vec{V} = \nabla \phi$

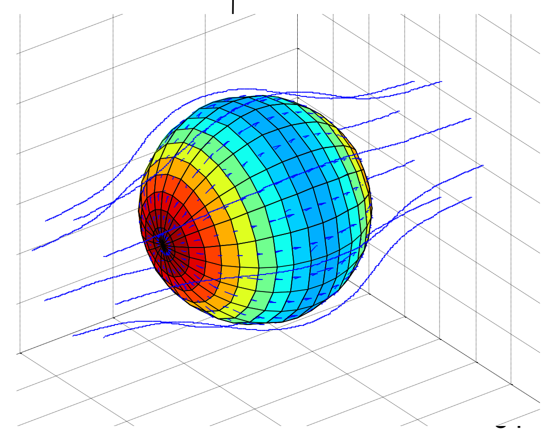
CONTINUITY : $\vec{\nabla} \cdot \vec{V} = 0 \Rightarrow \nabla^2 \phi = 0$

BERNOULLI'S : $C_p = 1 - \frac{|V|^2}{V_\infty^2}$



ELEMENTARY FLOWS :-

- UNIFORM FLOW
- POINT
- FILAMENTS
- PANELS



Uniform Flow

$$\vec{V} = \vec{V}_\infty = \text{Const} = u_\infty \hat{i} + v_\infty \hat{j} + w_\infty \hat{k}$$

$$\phi = \vec{V}_\infty \cdot \vec{r}$$

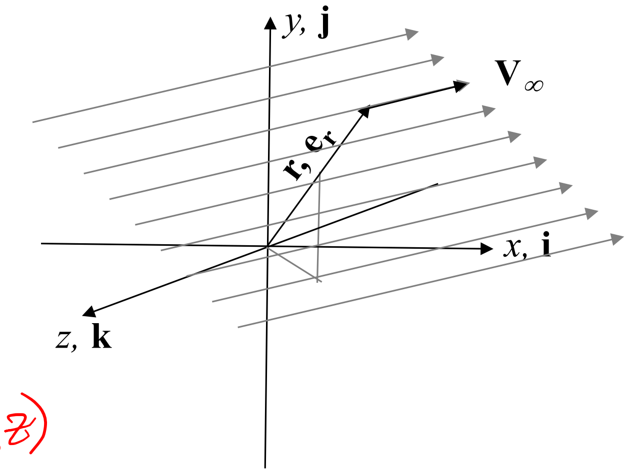
$$\vec{V} = \nabla\phi = \nabla\left(\vec{V}_\infty \cdot \vec{r}\right)$$

$$= \nabla\left(\left(u_\infty \vec{i} + v_\infty \vec{j} + w_\infty \vec{k}\right) \cdot \left(x\vec{i} + y\vec{j} + z\vec{k}\right)\right)$$

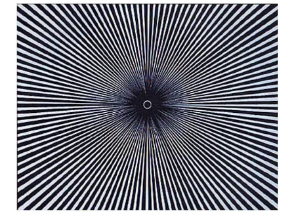
$$= \nabla\left(u_\infty x + v_\infty y + w_\infty z\right)$$

$$= \left(\frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j} + \frac{\partial}{\partial z}k\right)\left(u_\infty x + v_\infty y + w_\infty z\right)$$

$$= u_\infty \vec{i} + v_\infty \vec{j} + w_\infty \vec{k} = \vec{V}_\infty$$



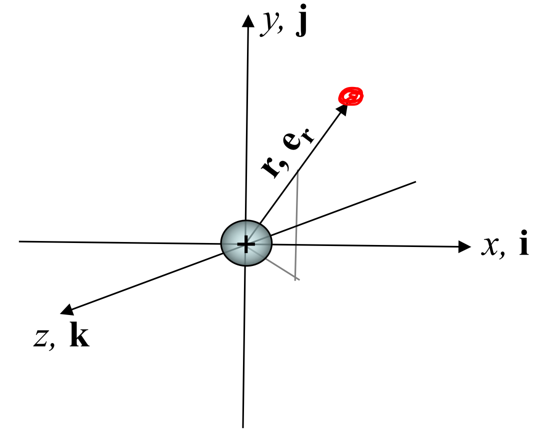
Point Source/Sink



1) AT ORIGIN:

$$\vec{V} = \frac{q}{4\pi r^2}\cdot \hat{e}_r \qquad \phi = \frac{-q}{4\pi r}$$

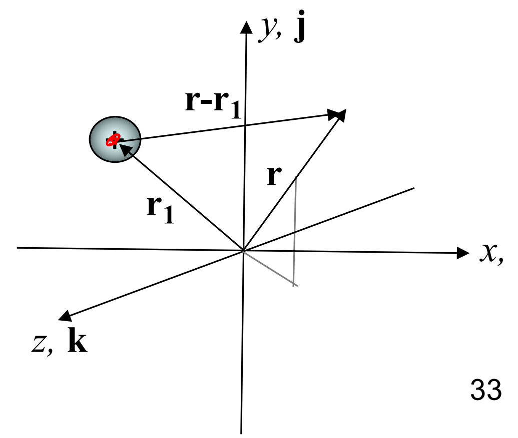
q = VOLUMETRIC FLOW RATE

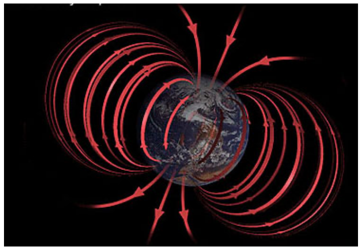


2) At $\vec{r}_1$

$$\phi = \frac{-q}{4\pi |r - r_1|}$$

$$\vec{V} = \frac{q(\vec{r} - \vec{r}_1)}{4\pi |r - r_1|^3}$$





Point Doublet

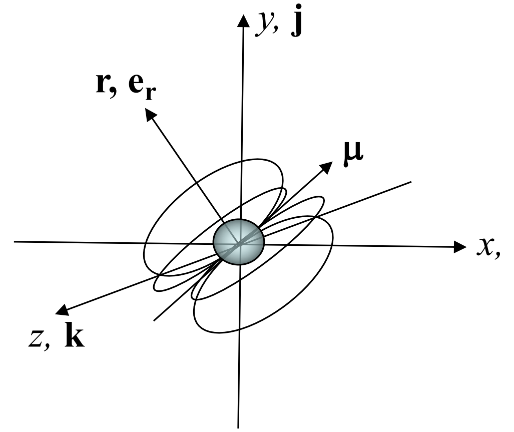
1) AT ORIGIN :

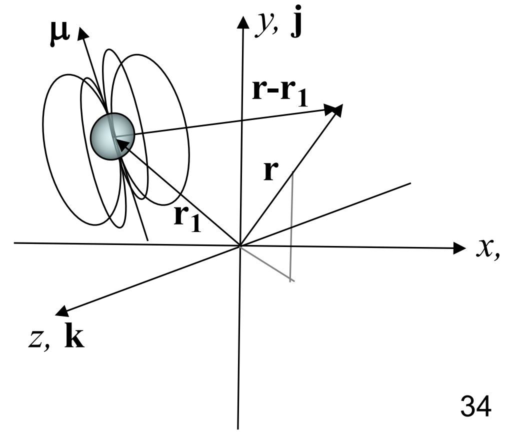
$$\phi = \frac{-\vec{\mu} \cdot \vec{e_r}}{4\pi r^2}$$

$$\vec{V} = \nabla \phi = \frac{\vec{\mu} \cdot \vec{e_r}}{2\pi r^3} - \frac{\vec{\mu} \cdot \nabla \vec{e_r}}{4\pi r^2}$$

2) At $\vec{r_1}$

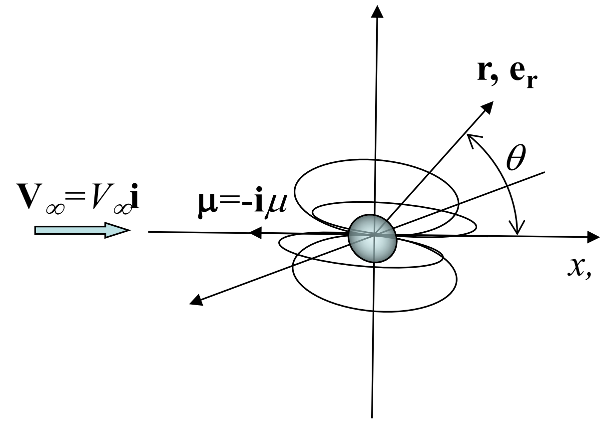
$$\phi = \frac{-\vec{\mu} \cdot (\vec{r} - \vec{r_1})}{4\pi \, |\vec{r} - \vec{r_1}|^3}$$





Flow Past a Sphere

Uniform flow + opposing doublet



$$\mathbf{V} = \mathbf{V}_{\infty}, \qquad \mathbf{V} = -\frac{\boldsymbol{\mu}.\mathbf{e}_r}{2\pi r^3}\mathbf{e}_r - \frac{\boldsymbol{\mu}.\nabla\mathbf{e}_r}{4\pi r^2}$$

$$\vec{V} = V_\infty \vec{i} - \frac{\vec{i}\mu \cdot \vec{e}_r}{2\pi r^3}\vec{e}_r - \frac{-\mu\vec{i}\cdot\nabla\vec{e}_r}{4\pi r^2}$$

$$\vec{V} = V_\infty \vec{i} - \frac{\mu\cos\theta}{2\pi r^3}\vec{e}_r - \frac{\mu\sin\theta}{4\pi r^3}\vec{e}_\theta$$

STAGNATION AT: $\theta = 0, \pi$

$$r = \sqrt[3]{\frac{\mu}{2\pi V_\infty}} = \text{sphere radius}$$

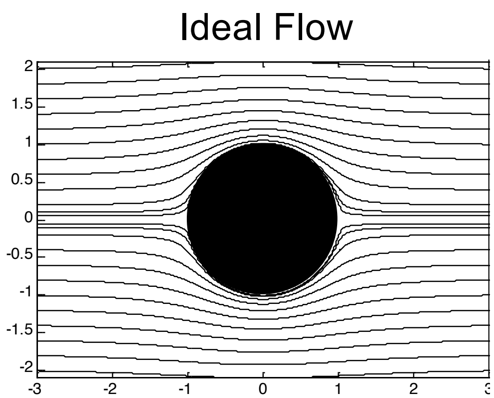
SO, ON SPHERE

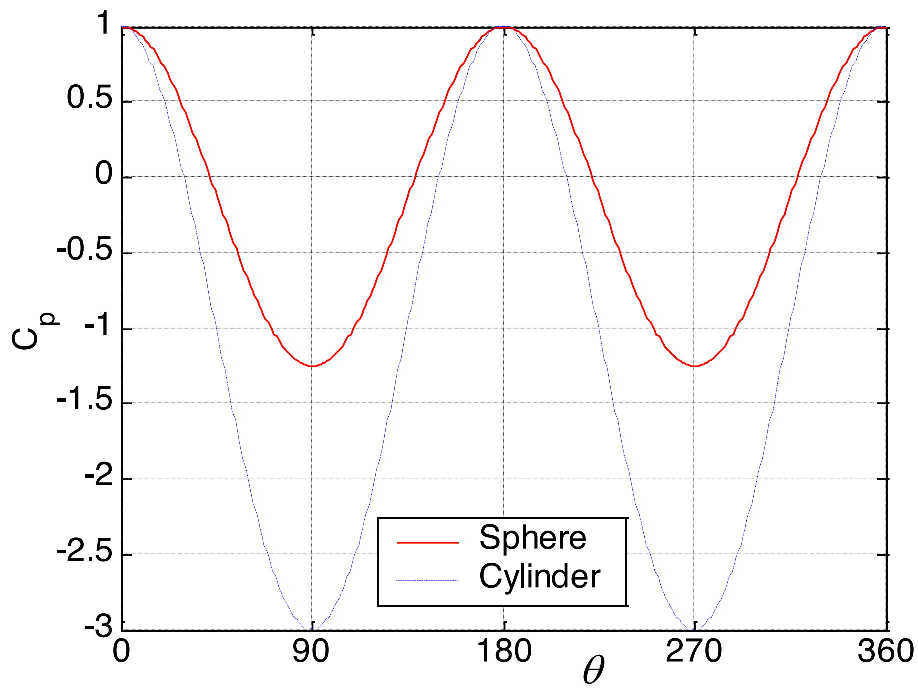
$$\vec{V} = V_\infty \vec{i} - V_\infty\cos\theta\,\vec{e}_r - \frac{1}{2}V_\infty\sin\theta\,\vec{e}_\theta$$

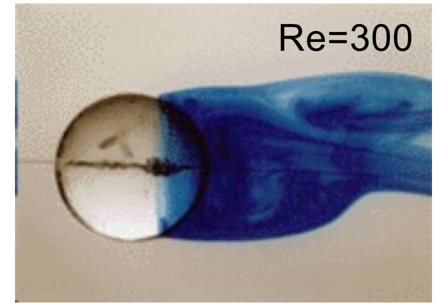
$$C_p = 1 - \frac{|V|^2}{V_\infty^2} = 1 - \frac{9}{4}\sin^2\theta$$

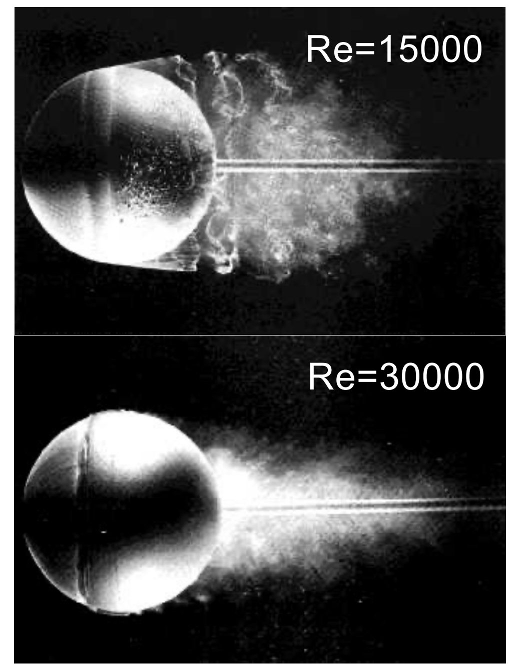
Flow Past a Sphere

Uniform flow + opposing doublet

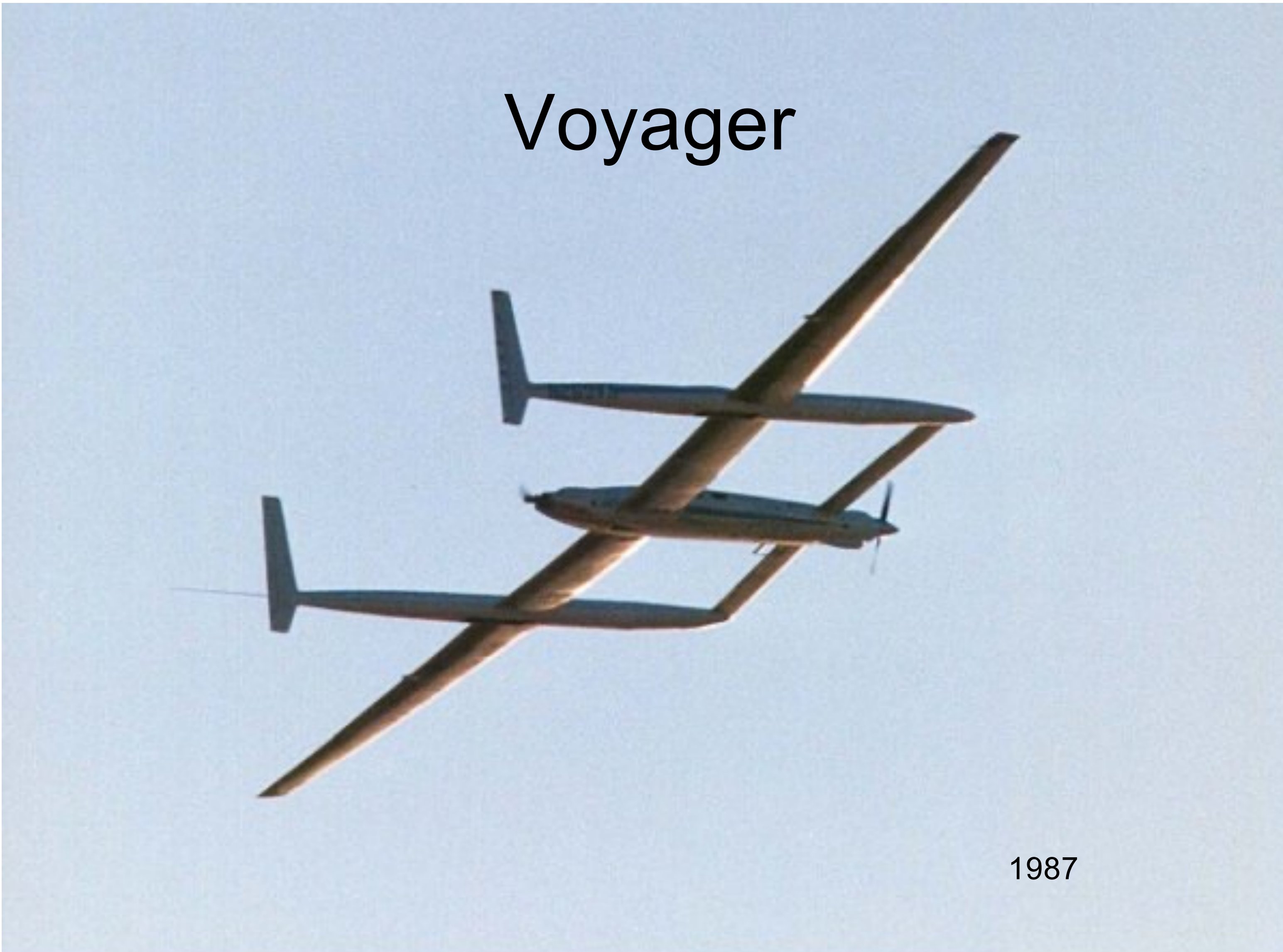








ONERA photographs, Werle 1980


Voyager
1987

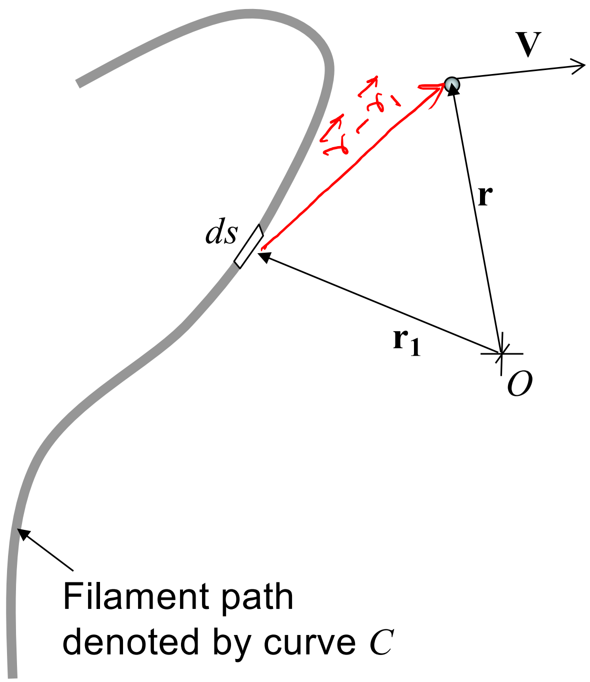
Point source
$\cancel{\varphi} = -\frac{q}{4\pi|\mathbf{r}-\mathbf{r}_1|}$

Lines/Filaments

Source, doublet or vortex distributed along a curved or straight line – source and doublet analysis analogous to panel

E.G : SOURCE FILAMENT

$d\phi(\vec{r}) = -\frac{q(\vec{r}_1)\cdot ds}{4\pi|r-r_1|}$

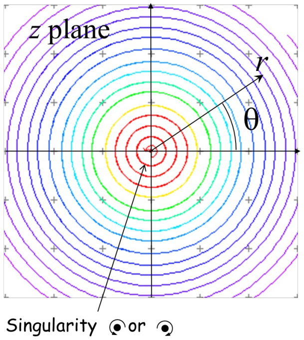


Filament path denoted by curve C

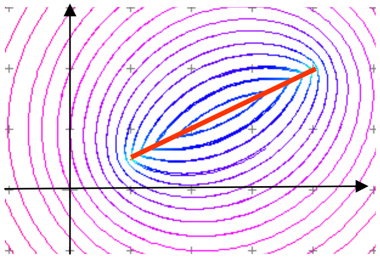
SO FOR WHOLE FILAMENT

$\phi(r) = -\frac{1}{4\pi}\oint\frac{q(\vec{r}_1)ds}{|\vec{r}-\vec{r}_1|}$

Helmholtz' Vortex Theorems Part 1

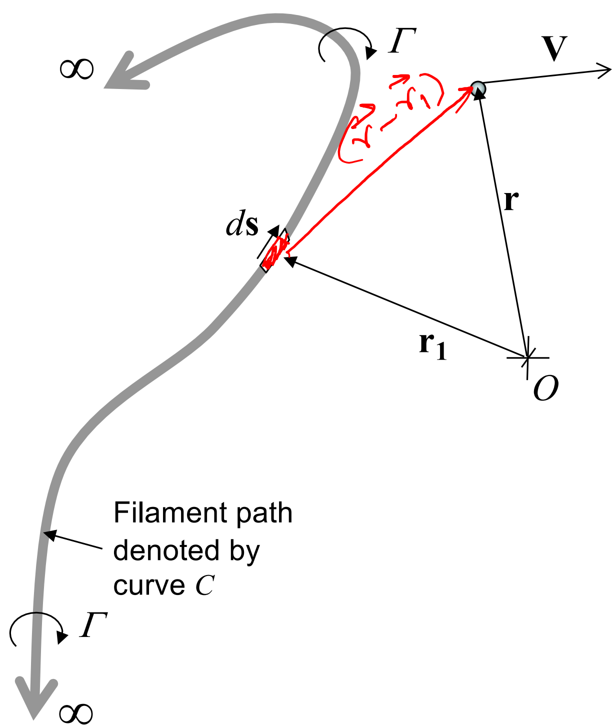


- The strength of a vortex tube (defined as the circulation around it) is constant along the tube.
- The tube, and the vortex lines from which it is composed, can therefore never end. They must extend to infinity or form loops.
- The average vorticity magnitude inside a vortex tube is inversely proportional to the cross-sectional area of the tube



Vortex Filament

Must obey Helmholtz's Laws so filament strength Γ must be constant, and filament must be looped or extend to infinity



Velocity field is determined from the ***Biot Savart Law***:

$$\mathbf{V}(\mathbf{r}) = -\frac{\Gamma}{4\pi}\int_C \frac{(\mathbf{r}-\mathbf{r}_1)\times d\mathbf{s}(\mathbf{r}_1)}{|\mathbf{r}-\mathbf{r}_1|^3}$$

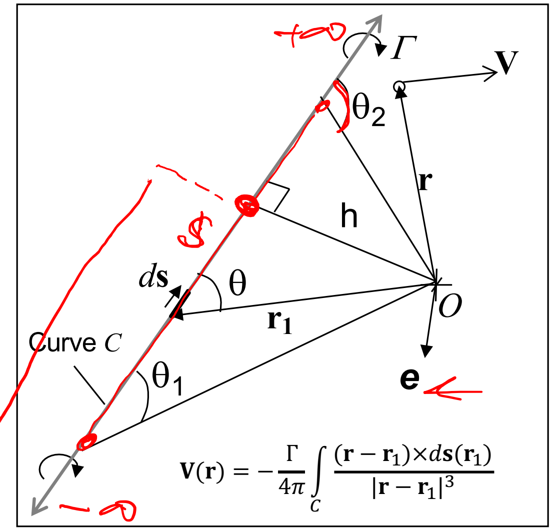
The Biot Savart Law cannot be inferred from simple integration since there is no comparable point singularity. Instead it is determined from considering the general problem of determining a velocity field from a given vorticity field.

$$\vec{\Omega} = \vec{\nabla}\times\vec{V}$$

$$\downarrow$$

$$\vec{V} = f(\vec{\Omega})$$

Example: Velocity induced by a section of a straight filament



•) PLACE ORIGIN AT POINT OF INTEREST $\vec{r} = 0$

$$\vec{V} = +\frac{\Gamma}{4\pi}\int_C \frac{+\vec{r}_1 \times d\vec{s}}{|\vec{r}_1|^3}$$

$$r_1 = h/\sin\theta$$

$$\vec{r}_1 \times d\vec{s} = |h/\sin\theta . ds . \sin(\theta)|\vec{e} = |h\, ds|\vec{e}$$

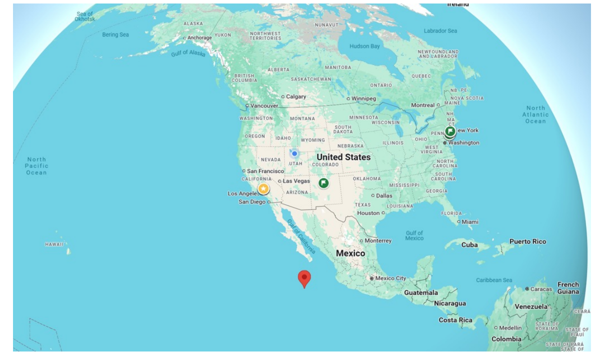
Now, $s = h/\tan\theta \Rightarrow ds = -h/\sin^2\theta \, d\theta$

$$\text{So, } \vec{V} = \frac{\Gamma\vec{e}}{4\pi}\int_{\theta_1}^{\theta_2} \frac{h^2/\sin^2\theta}{h^3/\sin^3\theta} = \frac{\Gamma\vec{e}}{4\pi h}\int_{\theta_1}^{\theta_2} \sin\theta \, d\theta = \frac{\Gamma\vec{e}}{4\pi h}(\cos\theta_1 - \cos\theta_2)$$

For semi-infinite : $\vec{V} = \dfrac{\Gamma\vec{e}}{4\pi h}$

For infinite filament : $\vec{V} = \Gamma/2\pi h \; \vec{e}$


Smithsonian
CHANNEL


United States
Mexico

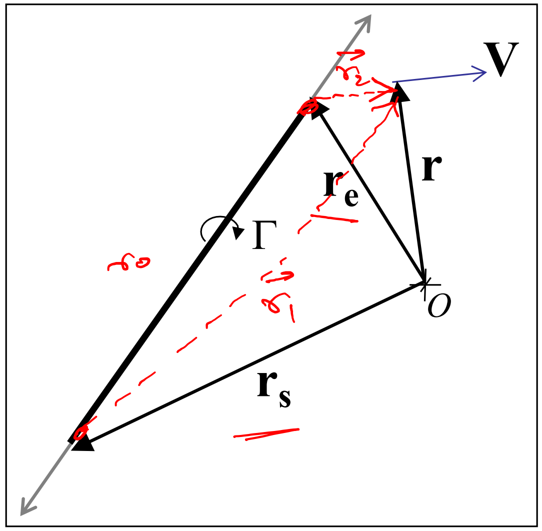
Example: Velocity induced by a section of a straight filament

So :- If $\vec{r}_1 = \vec{r}_s - \vec{r}$, $\vec{r}_2 = \vec{r}_e - \vec{r}$, $\vec{r}_0 = \bar{r}_1 - \bar{r}_2$

$$\vec{V} = \frac{\Gamma}{4\pi} \cdot \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|^2} \cdot \vec{r}_0 \cdot \left(\frac{\vec{r}_1}{r_1} - \frac{\vec{r}_2}{r_2} \right)$$

IN FUNCTIONAL FORM :

$$\vec{V} = \Gamma \cdot f_{fil} \left(\vec{r}, \vec{r}_s, \vec{r}_e \right)$$

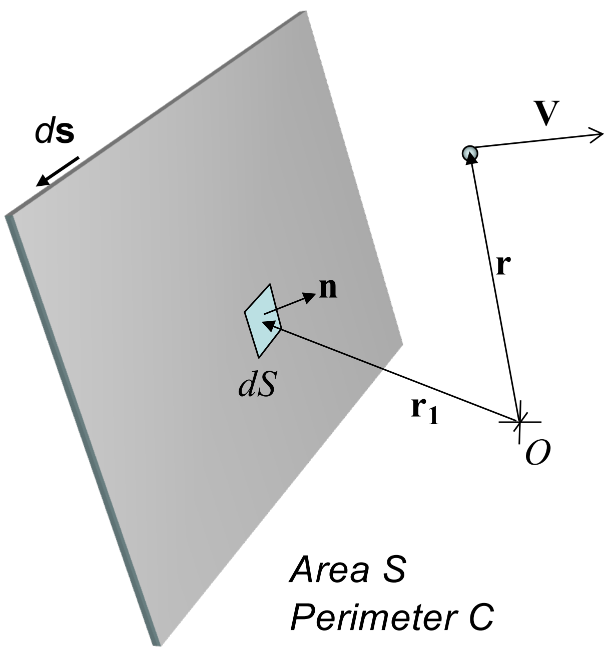


Point doublet

$$\varphi = -\frac{\boldsymbol{\mu}.(\mathbf{r}-\mathbf{r}_1)}{4\pi|\mathbf{r}-\mathbf{r}_1|^3}$$

Panels

Source or doublet distributed over finite (often flat) sheet or 'panel'



Area S
Perimeter C

E.G: DOUBLET PANEL: EACH ELEMENT BEHAVES AS AN OUTWARD POINTING DOUBLET.

$$d\phi = -\frac{\mu(\vec{r}_1)\vec{n}\cdot(\vec{r}-\vec{r}_1)\,dS}{4\pi|\vec{r}-\vec{r}_1|^3}$$

WHOLE PANEL:

$$\phi = \int_S \text{———} \;''\; \text{———}$$

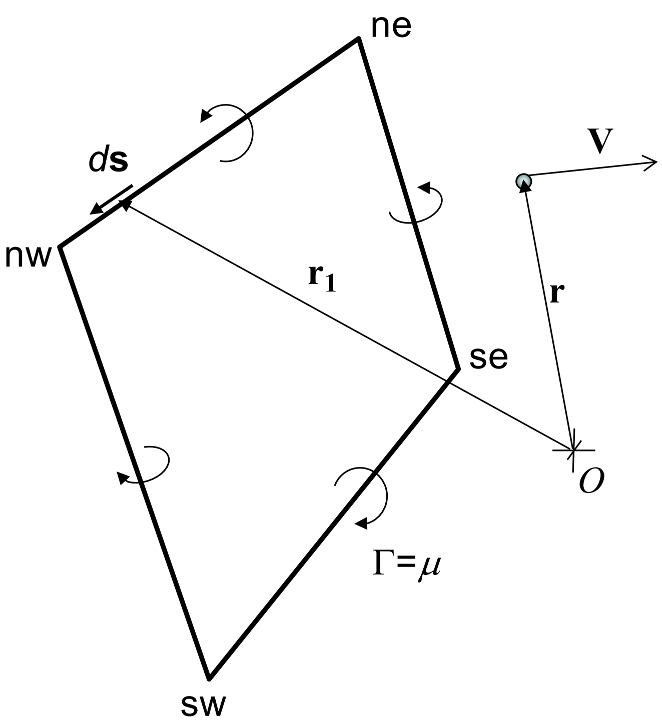
CONSTANT STRENGTH PANEL: $\mu(\vec{r}_1) = \mu$

EVALUATE $\nabla\phi$ for velocity:

$$\vec{V} = -\frac{\mu}{4\pi}\int\frac{(\vec{r}-\vec{r}_1)\times ds(r_1)}{|\vec{r}-\vec{r}_1|^3}$$

Constant Strength Doublet Panel

Same flow as a vortex filament ring around the panel perimeter



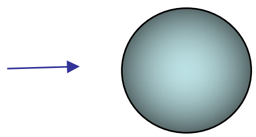
$$\mathbf{V}(\mathbf{r}) = -\frac{\Gamma}{4\pi}\oint_C \frac{(\mathbf{r}-\mathbf{r}_1)\times d\mathbf{s}(\mathbf{r}_1)}{|\mathbf{r}-\mathbf{r}_1|^3}$$

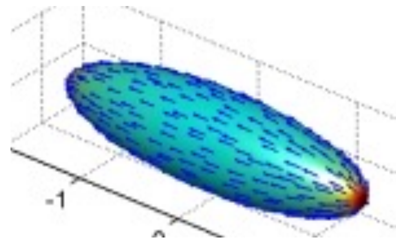
Therefore for a quadrilateral panel,

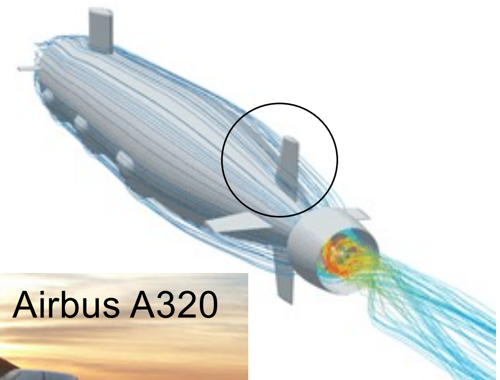
$$\mathbf{V}(\mathbf{r}) = \big[\mathbf{f}_{fil}(\mathbf{r},\mathbf{r}_{nw},\mathbf{r}_{sw}) + \mathbf{f}_{fil}(\mathbf{r},\mathbf{r}_{sw},\mathbf{r}_{se}) + \mathbf{f}_{fil}(\mathbf{r},\mathbf{r}_{se},\mathbf{r}_{ne}) + \mathbf{f}_{fil}(\mathbf{r},\mathbf{r}_{ne},\mathbf{r}_{nw})\big]\Gamma$$

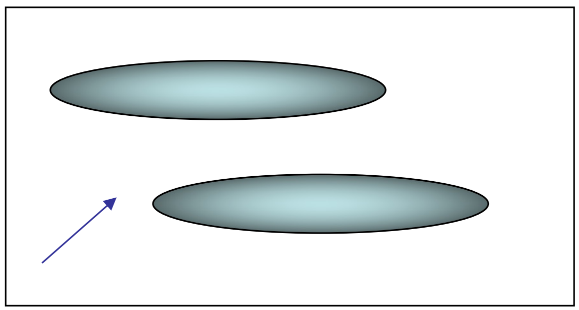
This makes doublet / vortex ring panels ideal for 3D panel methods, since their velocity fields are easy to compute. Since they contain a vortex element they can also be extended to situations (like wings) where vorticity is shed into the wake.

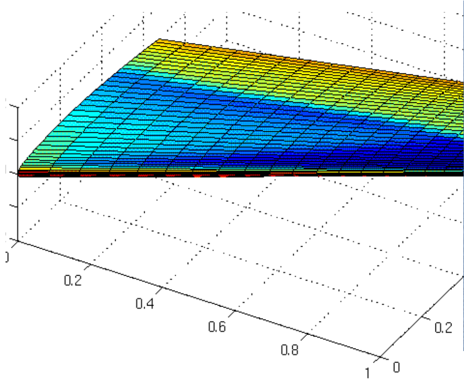
3D Flow





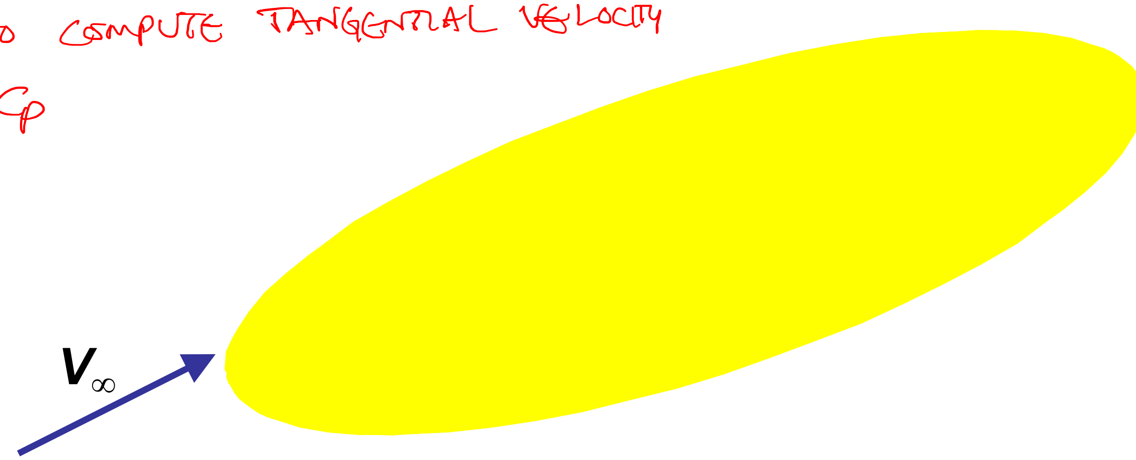






3D Doublet Panel Method

1) COVER BODY WITH N DOUBLET PANELS WITH N UNKNOWN STRENGTHS
2) PLACE A CONTROL POINT AT CENTER OF EACH PANEL
3) WRITE AN EXPRESSION DOWN FOR THE NORMAL COMPONENT OF VELOCITY AT EACH CONTROL POINT PRODUCED BY FREE-STREAM + THE N PANELS
4) SOLVE THESE N EQUATIONS FOR N PANEL STRENGTHS
5) USE CALCULATED STRENGTHS TO COMPUTE TANGENTIAL VELOCITY AT CONTROL POINTS AND Cp



3D Doublet Panel Code

Non-lifting bodies

- Handling vectors in Matlab
- Specifying a 3D body
- Specifying Panel Geometry
- Panel Influence – solving for the panel strengths
- Getting the surface pressure

```
%3D doublet panel method for acyclic flow around 3D bodies.
clear all;
vinf=[1;0;0]; %free stream velocity
```

Handling vectors in Matlab

```
%Specify body of revolution geometry
th=-pi/2:pi/20:pi/2;xp=sin(th);yp=cos(th);nc=20;
[r,rc,nw,sw,se,ne,no,we,so,ea]=bodyOfRevolution(xp,yp,nc);
```

Specifying a 3D body, the panel geometry, and indexing

```
% determine surface area and normal vectors at control points (assumes counter clockwise around compass by RH rule points out c
ac=0.5*v_cross(r(:,sw)-r(:,ne),r(:,se)-r(:,nw));nc=ac./v_mag(ac);
```

Panel influence…

```
%determine influence coefficient matrix coeff
npanels=length(rc(1,:));coef=zeros(npanels);
for n=1:npanels
    cmn=ffil(rc(:,n),r(:,nw),r(:,sw))+ffil(rc(:,n),r(:,sw),r(:,se))+ffil(rc(:,n),r(:,se),r(:,ne))+ffil(rc(:,n),r(:,ne),r(:,nw))
    coef(n,:)=nc(1,n)*cmn(1,:)+nc(2,n)*cmn(2,:)+nc(3,n)*cmn(3,:);
end

%determine result vector and solve matrix for filament strengths
rm=(-nc(1,:)*vinf(1)-nc(2,:)*vinf(2)-nc(3,:)*vinf(3))';
coef(end+1,:)=1;rm(end+1)=0; %prevents singular matrix - sum of panel strengths on closed body is zero

ga=coef\rm;
```

… and solving for the panel strengths

Getting the surface pressure

```
%Determine velocity and pressure at control points
ga=repmat(ga',[3 1]);
for n=1:npanels     %Determine velocity at each c.p. without principal value
    cmn=ffil(rc(:,n),r(:,nw),r(:,sw))+ffil(rc(:,n),r(:,sw),r(:,se))+ffil(rc(:,n),r(:,se),r(:,ne))+ffil(rc(:,n),r(:,ne),r(:,nw))
    v(:,n)=vinf+sum(ga.*cmn,2);
end                 %Determine principle value of velocity at each c.p., -grad(ga)/2
gg=v_cross((rc(:,we)-rc(:,no)).*(ga(:,we)+ga(:,no))+(rc(:,so)-rc(:,we)).*(ga(:,so)+ga(:,we))+(rc(:,ea)-rc(:,so)).*(ga(:,ea)+ga(

v=v-gg/2; %velocity vector
cp=1-sum(v.^2)/(vinf'*vinf); %pressure
```

```
%plotting of pressure distribution and velocity vectors
```

Vectors In Matlab

Use arrays with 3 rows, one for each component. E.g. $\mathbf{V}_\infty$

[u; v; w]

```
%3D doublet panel method for flow around 3D bodies.
clear all;
vinf=[1;0;0]; %free stream velocity
```

Easy to make functions for vector operations like dot and cross products, and magnitude

```
function c=v_dot(a,b);
c=zeros(size(a));
c(1,:)=a(1,:).*b(1,:)+a(2,:).*b(2,:)+a(3,:).*b(3,:);
c(2,:)=c(1,:);c(3,:)=c(1,:);
```

```
function c=v_cross(a,b);
c=zeros(size(a));
c(1,:)=(a(2,:).*b(3,:)-a(3,:).*b(2,:));
c(2,:)=(a(3,:).*b(1,:)-a(1,:).*b(3,:));
c(3,:)=(a(1,:).*b(2,:)-a(2,:).*b(1,:));
```

With these it is now easier to make more complex functions

E.g. $\mathbf{f}_{fil}(\mathbf{r},\mathbf{r}_s,\mathbf{r}_e)$

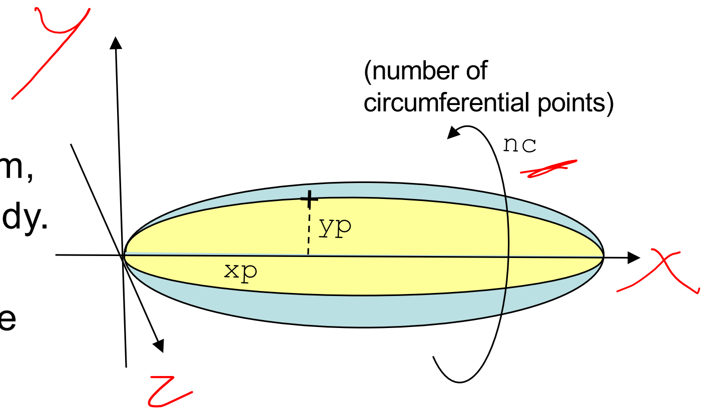
```
function q=ffil(r,rs,re);
r1(1,:)=rs(1,:)-r(1,:); r1(2,:)=rs(2,:)-r(2,:); r1(3,:)=rs(3,:)-r(3,:);
r2(1,:)=re(1,:)-r(1,:); r2(2,:)=re(2,:)-r(2,:); r2(3,:)=re(3,:)-r(3,:);
r0(1,:)=r1(1,:)-r2(1,:);r0(2,:)=r1(2,:)-r2(2,:);r0(3,:)=r1(3,:)-r2(3,:);
c=v_cross(r1,r2);
c2=v_dot(c,c);
q=c./c2.*v_dot(r0,r1./v_mag(r1)-r2./v_mag(r2))/4/pi;
```

Specifying a 3D Body

First we must choose shape. E.g. Body of revolution:

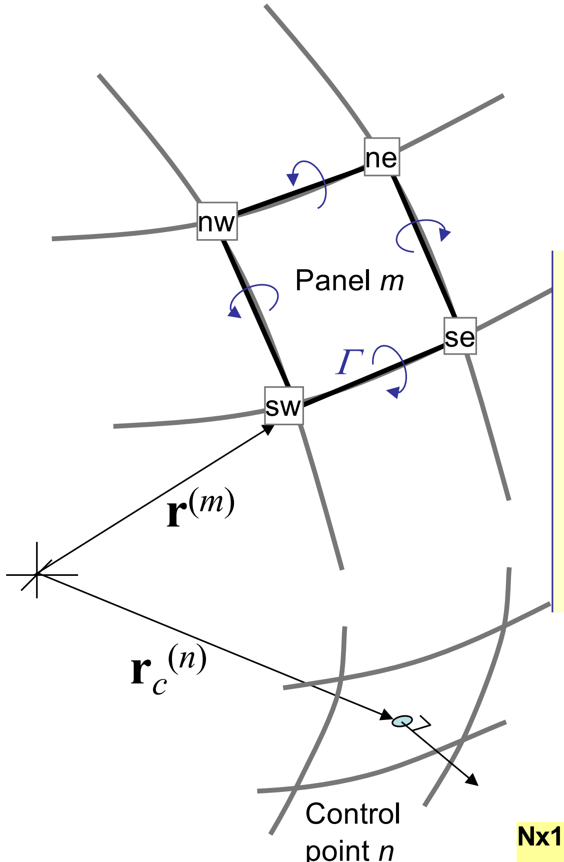
```
%Specify body of revolution geometry
th=-pi/2:pi/20:pi/2;xp=sin(th);yp=cos(th);nc=20;
[r,rc,nw,sw,se,ne,no,we,so,ea]=bodyOfRevolution(xp,yp,nc)
```

Then we generate the coordinates, in vector form, of all the points on the body. These points (in a rectangular array, become the panel corner points $\mathbf{r}$




```
function [r,rc,nw,sw,se,ne,no,we,so,ea]=bodyOfRevolution(xp,yp,nc)

%Define vertices of panels
x=repmat(xp,[nc+1 1]);     %grid of x points
al=[0:nc]/(nc)*2*pi;     %vector of circumferential angles (about x axis)
y=cos(al)'*yp;
z=sin(al)'*yp;
r=zeros([3 size(x)]);r(1,:)=x(:);r(2,:)=y(:);r(3,:)=z(:); %position vector of vertices
ri=reshape(1:prod(size(x)),size(x)); %index of vertices
```

Panel Influence



Velocity due to panel *m* at control point *n*:

$$\mathbf{V}(\mathbf{r}_c^{(n)})\Big|^{(m)} = \Big[\mathbf{f}_{fil}(\mathbf{r}_c^{(n)},\mathbf{r}_{nw}^{(m)},\mathbf{r}_{sw}^{(m)}) + \mathbf{f}_{fil}(\mathbf{r}_c^{(n)},\mathbf{r}_{sw}^{(m)},\mathbf{r}_{se}^{(m)}) + \mathbf{f}_{fil}(\mathbf{r}_c^{(n)},\mathbf{r}_{se}^{(m)},\mathbf{r}_{ne}^{(m)}) + \mathbf{f}_{fil}(\mathbf{r}_c^{(n)},\mathbf{r}_{ne}^{(m)},\mathbf{r}_{nw}^{(m)})\Big]\,\Gamma^{(m)}$$

```
%determine influence coefficient matrix coef
npanels=length(rc(1,:));coef=zeros(npanels);
for n=1:npanels
cmn=ffil(rc(:,n),r(:,nw),r(:,sw))+ffil(rc(:,n),r(:,sw),r(:,se))
   +ffil(rc(:,n),r(:,se),r(:,ne))+ffil(rc(:,n),r(:,ne),r(:,nw));
coef(n,:)=nc(1,n)*cmn(1,:)+nc(2,n)*cmn(2,:)+nc(3,n)*cmn(3,:);
end

%determine result vector and solve matrix for filament strengths
rm=(-nc(1,:)*vinf(1)-nc(2,:)*vinf(2)-nc(3,:)*vinf(3))';
coef(end+1,:)=1;rm(end+1)=0; %prevents singular matrix
ga=coef\rm;
```

So, to get the $\Gamma^{(m)}$'s we need to solve the simultaneous equations:

$$\underbrace{-\mathbf{V}_\infty\mathbf{n}_c^{(n)}}_{\text{Nx1 matrix of freestream components}} = \sum_m \underbrace{\mathbf{C}^{(n,m)}.\,\mathbf{n}_c^{(n)}}_{\text{NxN coefficient matrix}}\underbrace{\Gamma^{(m)}}_{\text{Nx1 matrix of panel strengths}}$$

Nx1 matrix of freestream components

NxN coefficient matrix

Nx1 matrix of panel strengths

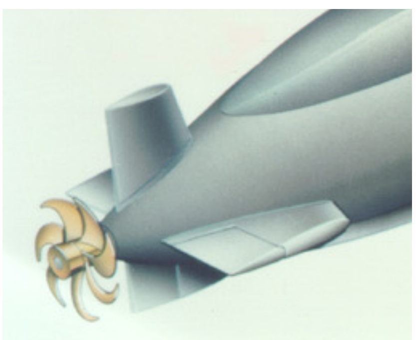
Using the Code

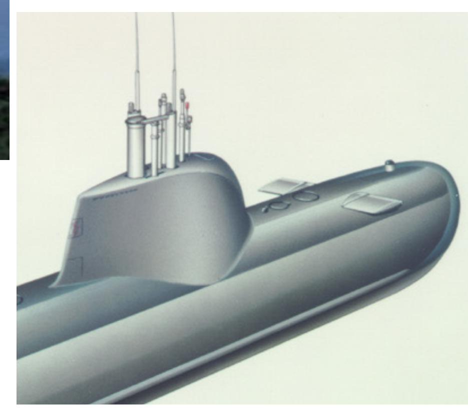
- Plotting the pressure, streamlines
- Deforming the body shape
- Changing the shape
- More than one body

Wings in Ideal Flow
or, How do Airplanes Fly?

1. The Physics of Lifting Surfaces
2. Wing Terminology, Forces and Moments
3. Ideal Flow Methods for Wings

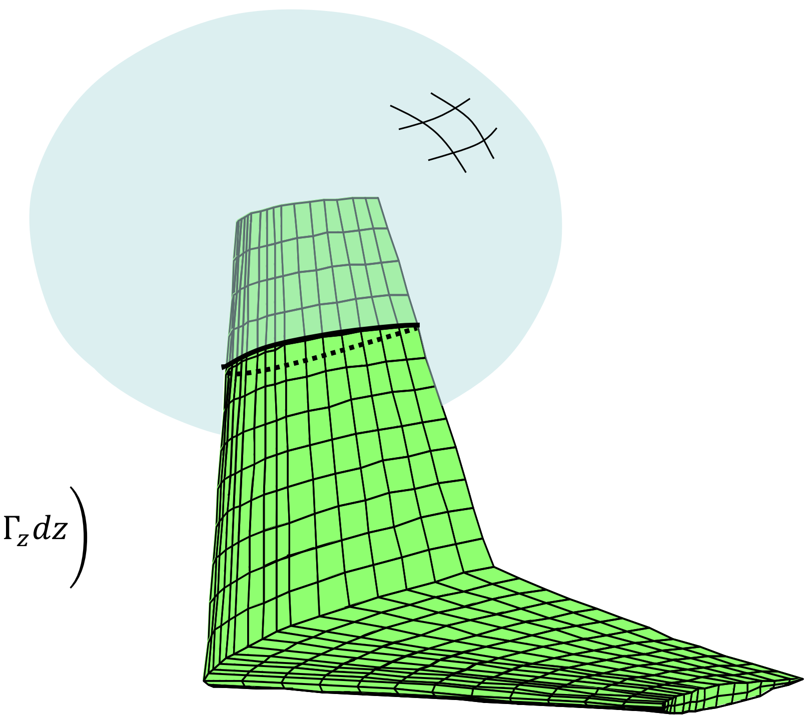
Wings





Why Airplanes Can't Fly

$$\int_S \nabla \times V.\mathbf{n}d\,S = \oint_C V.\mathrm{d}\mathbf{s} = \Gamma_C$$

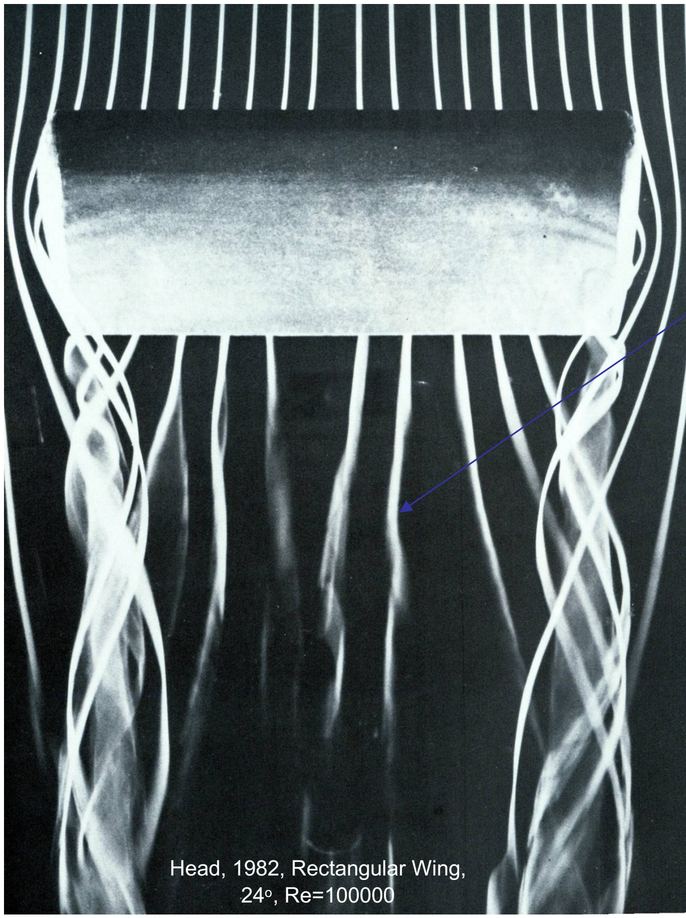


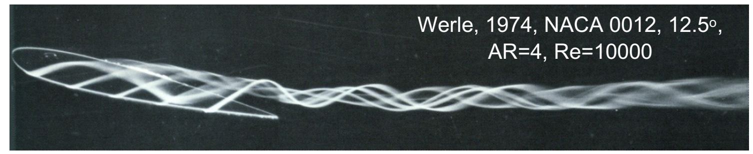
$$\mathbf{F} = -\rho \mathbf{U}_b \times \left(\mathbf{i} \int_{X\ body} \Gamma_x dx + \mathbf{j} \int_{Y\ body} \Gamma_y dy + \mathbf{k} \int_{Z\ body} \Gamma_z dz \right)$$

Boeing 727

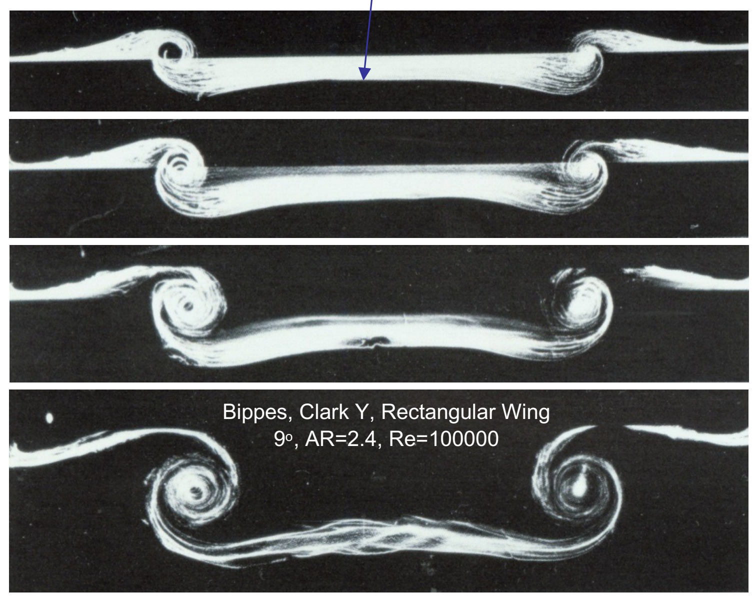
Dryden Flight Research Center ECN 3831 Photographed 1974
B-727 vortex study NASA photo

Physics of an Flow Over a Wing

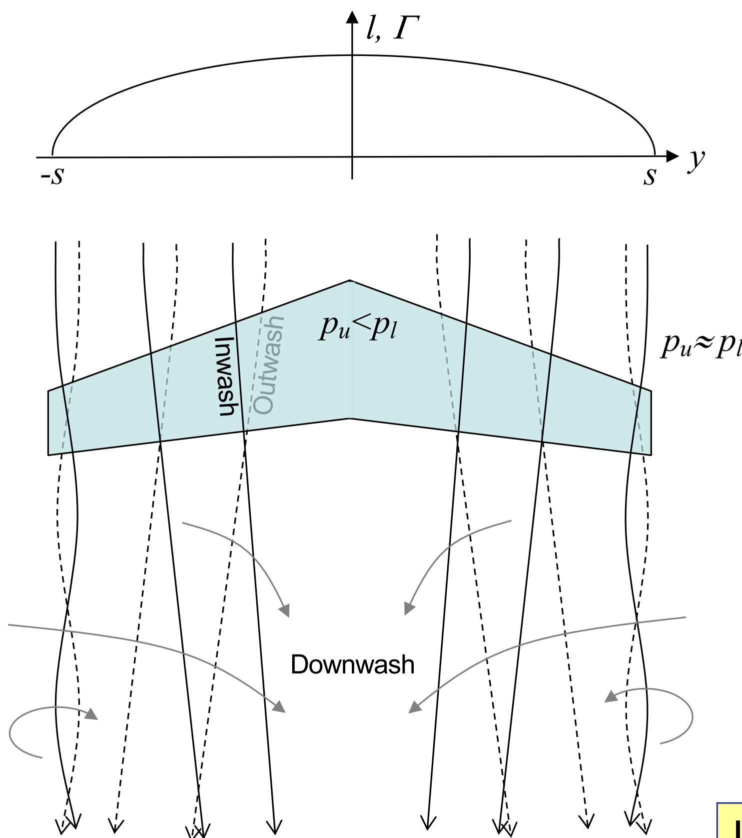

Head, 1982, Rectangular Wing, 24º, Re=100000


Werle, 1974, NACA 0012, 12.5º, AR=4, Re=10000

Wake descends roughly on an extension of the chord line


Bippes, Clark Y, Rectangular Wing 9º, AR=2.4, Re=100000

Physics of Flow Over a Wing



Lift varies across span

Circulation is shed (Helmholz thm)

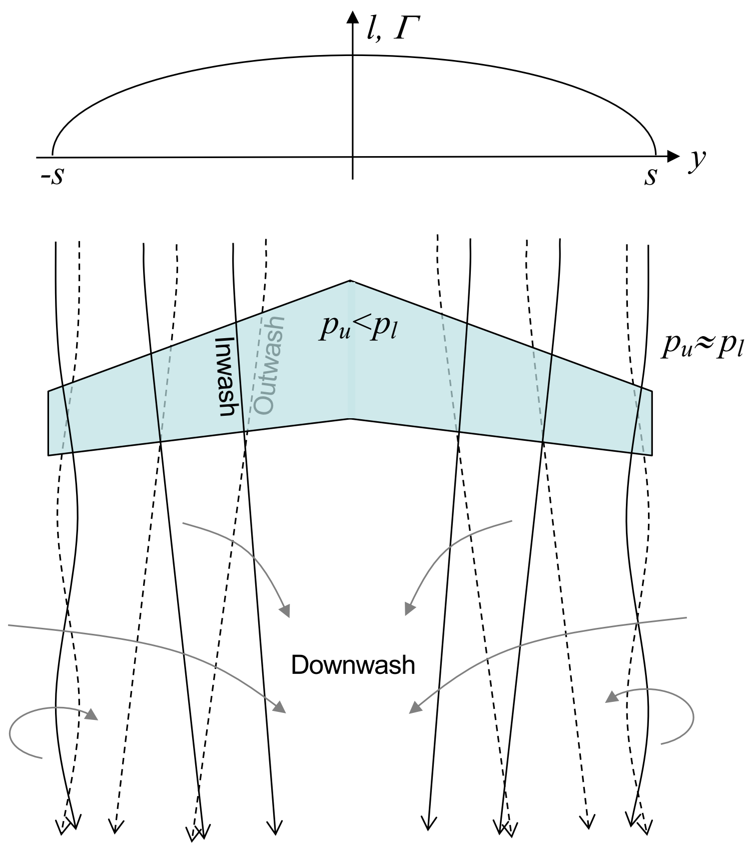
Vortical wake

Vortical wake induces downwash on wing…

…changing angle of attack just enough to produce variation of lift across span

Including a model of the vortical wake is thus *critical* to model a wing in ideal flow

Induced Drag



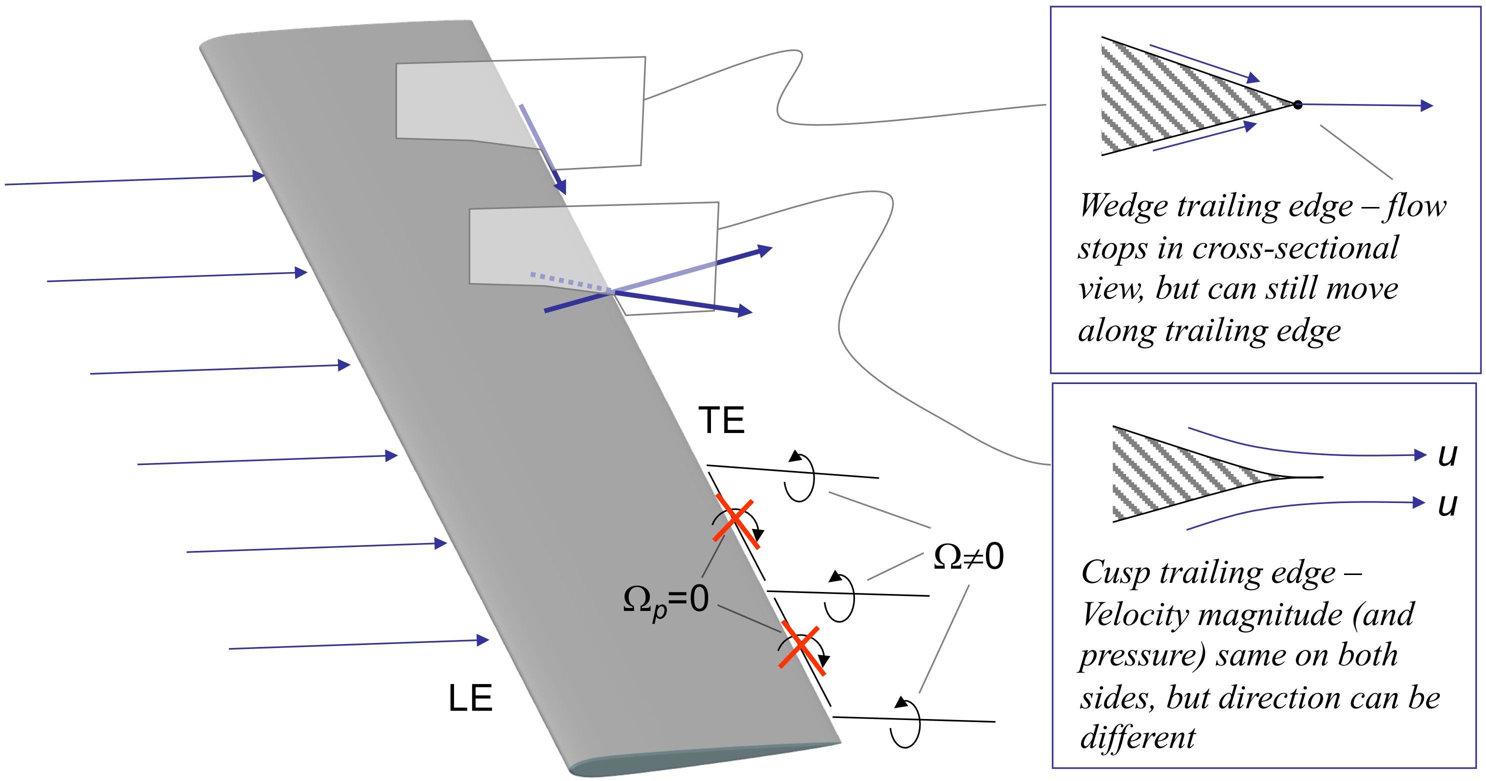
• The continuous generation of the vortical wake requires energy input (even in ideal flow)

• This energy is supplied by the wing doing work against an inviscid drag force.

• This *induced drag* (induced by the wake on the wing) is still consistent with ideal flow because, with the vortical wake, the wing is no longer an 'isolated body in an otherwise undisturbed ideal flow'.

• So, ideal flow pressures acting on a 3D wing shouldn't balance in the streamwise direction, when that wing is generating lift

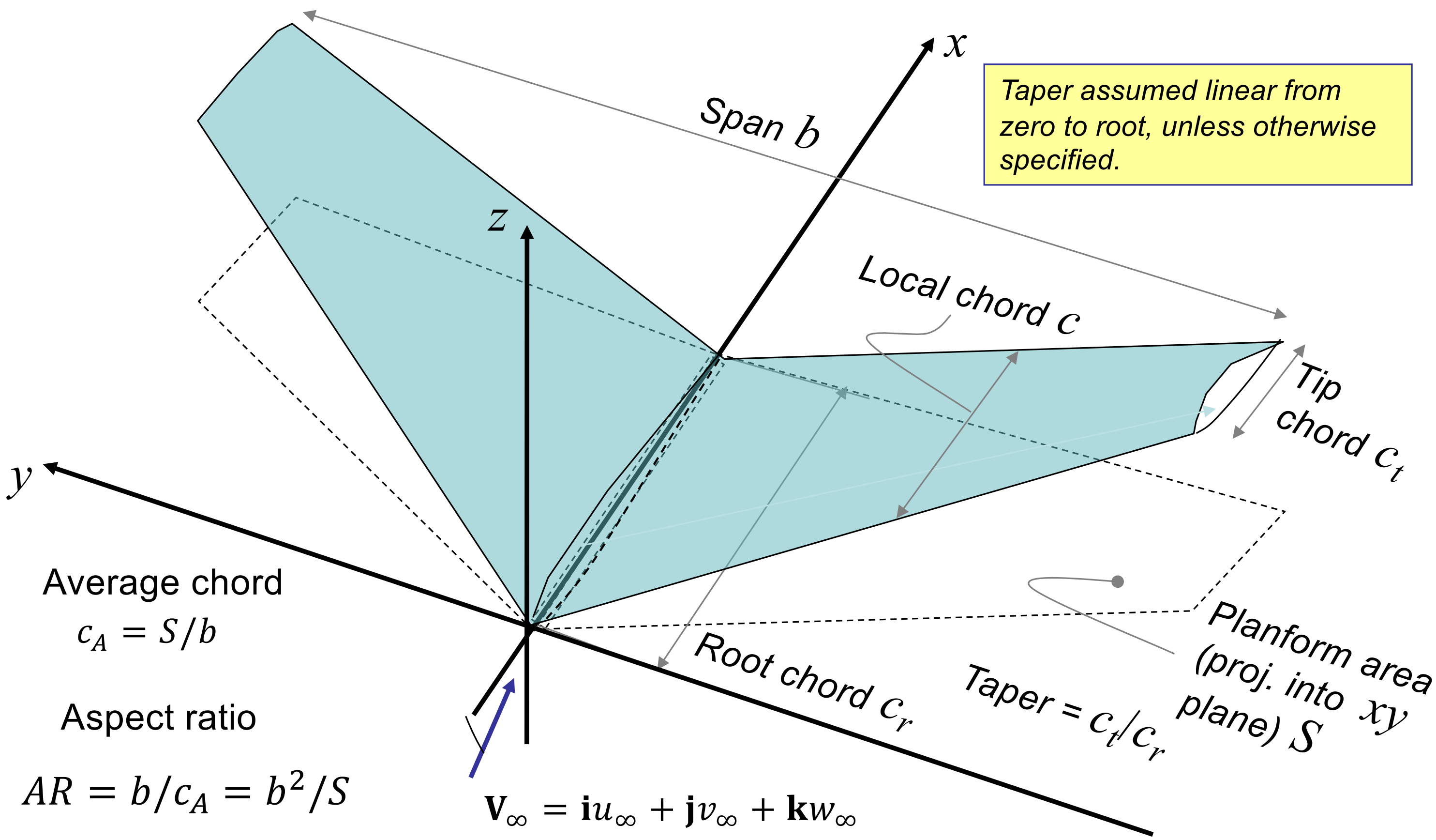
Kutta Condition for 3D



In both cases this is equivalent to requiring that the surface vorticity parallel to the t.e. Ω_p be zero. In practice, enforcing this will require (by Helmholtz' theorem) that streamwise vorticity to be shed from the trailing edge. This is how we get our vortical wake.

Wing Nomenclature

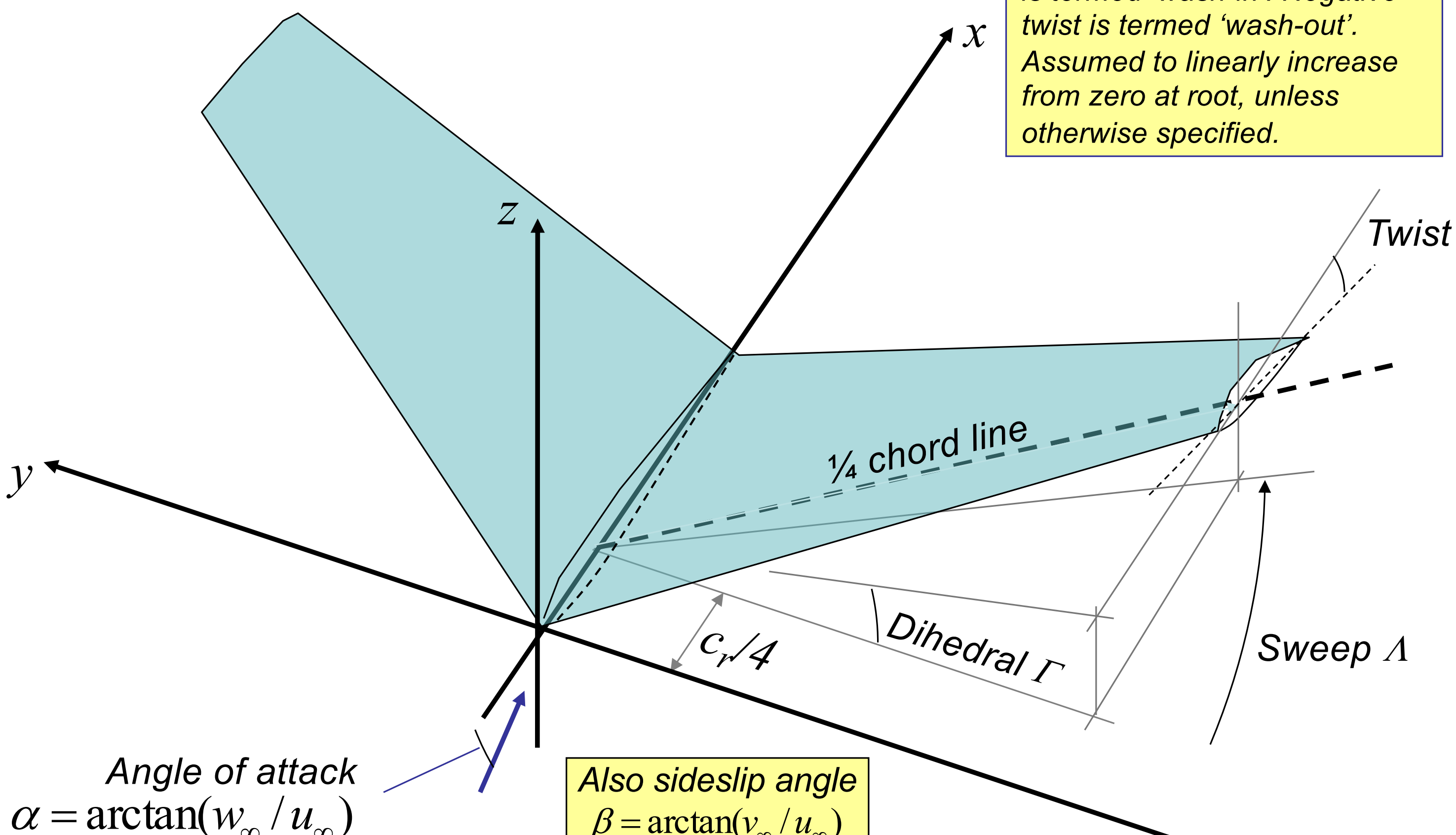
Areas and Lengths



Wing Nomenclature

Angles

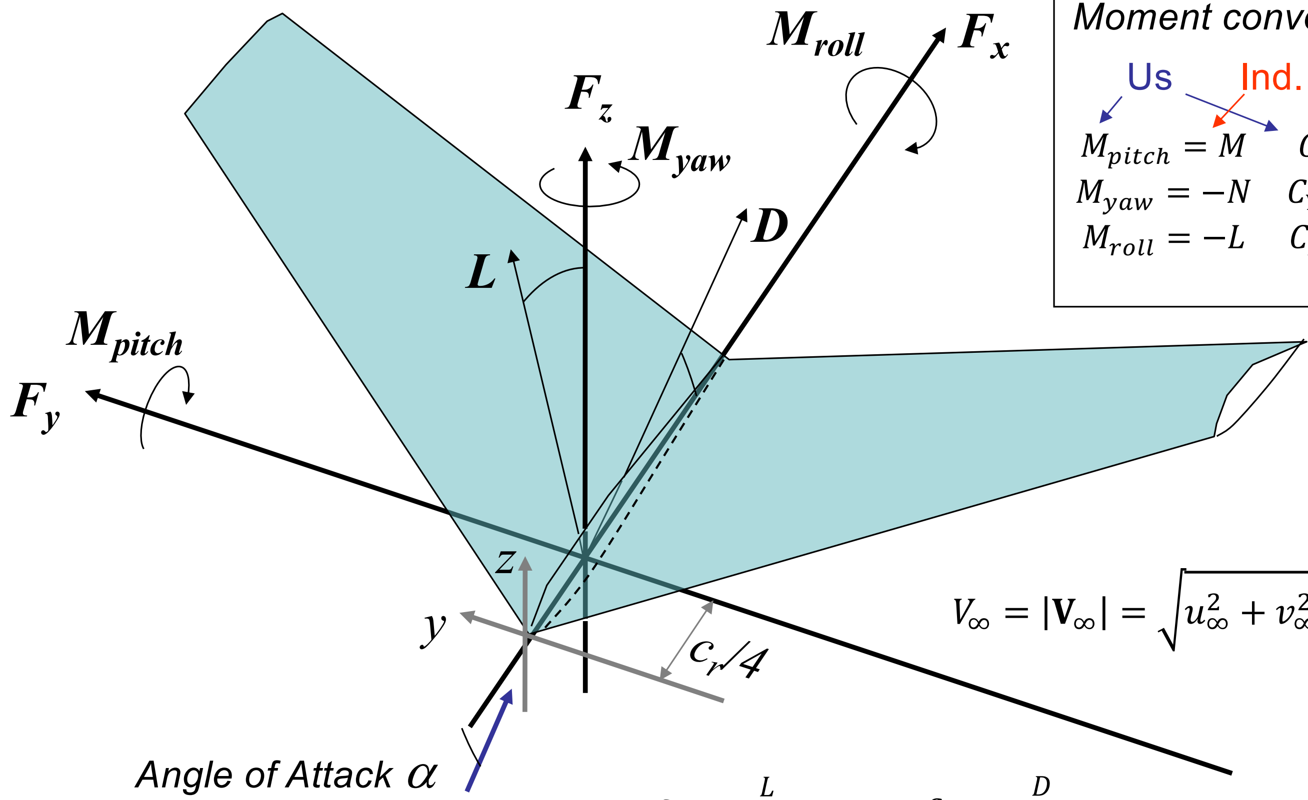
Positive twist increases α, and is termed 'wash-in'. Negative twist is termed 'wash-out'. Assumed to linearly increase from zero at root, unless otherwise specified.



Also sideslip angle
$\beta = \arctan(v_\infty / u_\infty)$
(zero as pictured)

Wing Nomenclature

Forces and Moments



Moment conventions

Us → M_{pitch}, C_P; Ind. std → M, C_M

$$M_{pitch} = M \qquad C_P = C_M$$
$$M_{yaw} = -N \qquad C_Y = -C_N$$
$$M_{roll} = -L \qquad C_R = -C_L$$

$$V_\infty = |\mathbf{V}_\infty| = \sqrt{u_\infty^2 + v_\infty^2 + w_\infty^2}$$

$$C_L = \frac{L}{\frac{1}{2}\rho V_\infty^2 S} \qquad C_D = \frac{D}{\frac{1}{2}\rho V_\infty^2 S} \qquad C_M = \frac{M}{\frac{1}{2}\rho V_\infty^2 S \bar{c}}$$

Wing Nomenclature

M

F_y

ions

$= C_M$

$-C_N$

$= -C_L$

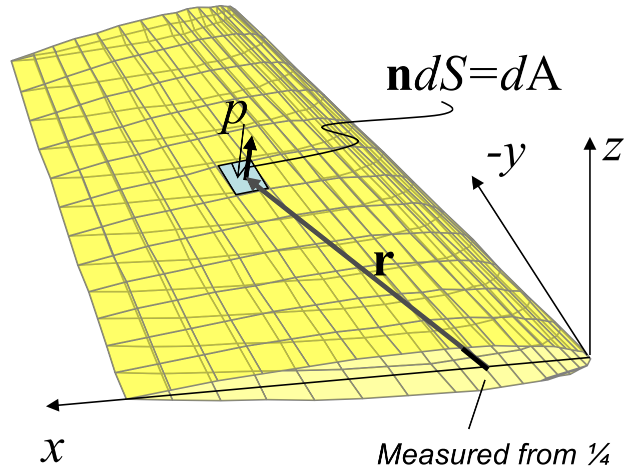
w_∞^2

$\frac{}{2}\rho V_\infty S\bar{c}$

Computing Forces and Moments

$$C_L = \frac{L}{\frac{1}{2}\rho V_\infty^2 S}$$

$$C_M = \frac{M}{\frac{1}{2}\rho V_\infty^2 S\bar{c}}$$



Measured from ¼ chord at root, so moments about this point

Planform area

$$S = \frac{1}{2}\oint_{Surface} |\mathbf{k}.\mathbf{n}|dS = \frac{1}{2}\oint_{Surface} |\mathbf{k}.d\mathbf{A}|$$

Total Force

$$\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k} = -\oint_{Surface} p\mathbf{n}dS = -\oint_{Surface} pd\mathbf{A}$$

as coefficients

$$C_\mathbf{F} = \frac{\mathbf{F}}{\frac{1}{2}\rho U_\infty^S S} = C_x\mathbf{i} + C_y\mathbf{j} + C_z\mathbf{k} = -\frac{1}{S}\oint_{Surface} C_p d\mathbf{A}$$

C_L and C_D obtained by rotating C_z and C_x by the angle of attack

Total moment about origin of **r**

$$\mathbf{M} = M_{roll}\mathbf{i} + F_{pitch}\mathbf{j} + F_{yaw}\mathbf{k} = -\oint_{Surface} p\mathbf{r}\times\mathbf{n}dS = -\oint_{Surface} p\mathbf{r}\times d\mathbf{A}$$

as coefficients

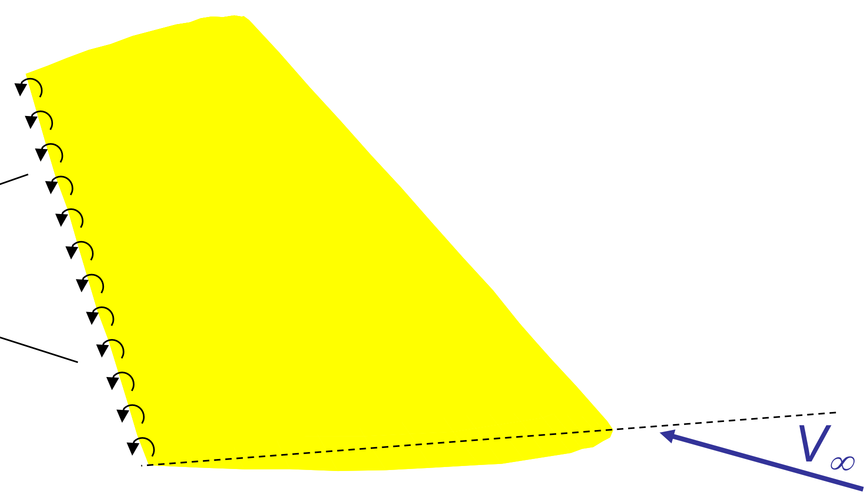
$$C_\mathbf{M} = \frac{\mathbf{M}}{\frac{1}{2}\rho U_\infty^S S\bar{c}} = C_R\mathbf{i} + C_P\mathbf{j} + C_Y\mathbf{k} = -\frac{1}{S\bar{c}}\oint_{Surface} C_p\mathbf{r}\times d\mathbf{A}$$

Doublet Panel Method for Wings

1. Treat wing like non-lifting body

• Cover wing with *N* panels. Determine panel strengths by requiring no flow through surface at *N* control points

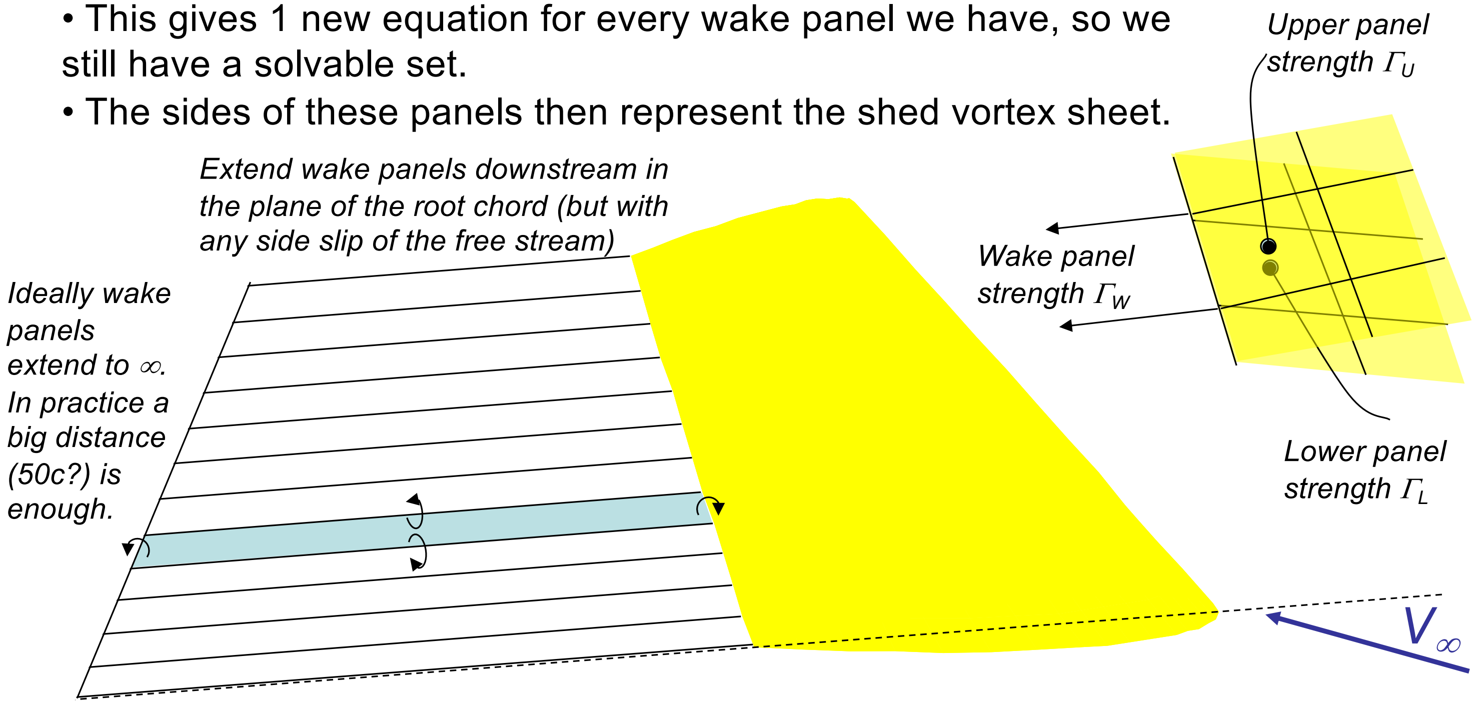
Solution will result in non zero filament strength along the trailing edge (Kutta condition not satisfied)



Doublet Panel Method for Wings

2. Add panels in wake to cancel t.e. vorticity

• The new wake panels have no control point requirements – we simply require that they cancel the t.e. vorticity so that $\Gamma_W = \Gamma_L - \Gamma_U$.
• This gives 1 new equation for every wake panel we have, so we still have a solvable set.
• The sides of these panels then represent the shed vortex sheet.



3D Wing Code

- Same as 3D Panel Code for non-lifting bodies except:
  - Specify different geometry
  - Extend influence coefficient matrix to include strength relation for each wake panel $\Gamma_W = \Gamma_L - \Gamma_U$, rather than a control point relation
  - Add code to compute forces and moments

```
%3D doublet panel method for lifting wings.
clear all;
vinf=[cos(8*pi/180);0;sin(8*pi/180)]; %free stream velocity
%Specify wing geometry (NACA 0012 section)
xp=[1   0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.25    0.2 0.15    0.1 0.075   0.05    0.025   0.0125  0   0.012                075
zp=[0   0.01448 0.02623 0.03664 0.04563 0.05294 0.05803 0.06002 0.05941 0.05737 0.05345 0.04683 0.042                01894
nb=25;b=2.5;sweep=30;dihedral=15;twist=10;taper=.5;
[r,rc,nw,sw,se,ne,no,we,so,ea,wp,bp]=wing(xp,zp,b,nb,vinf,sweep,dihedral,twist,taper);
xl=-1;xh=2;yl=-1.5;yh=1.5;zl=-1;zh=1;cl=-2;ch=1; % plotting limits

% determine surface area and outward pointing normal vectors at control points (assumes counter clockwise around compass by RH r
ac=0.5*v_cross(r(:,sw)-r(:,ne),r(:,se)-r(:,nw));nc=ac./v_mag(ac);

%determine influence coefficient matrix
npanels=length(rc(1,:));coef=zeros(npanels);
for nn=1:length(bp)
    n=bp(nn);
    cmn=ffil(rc(:,n),r(:,nw),r(:,sw))+ffil(rc(:,n),r(:,sw),r(:,se))+ffil(rc(:,n),r(:,se),r(:,ne))+ffi                ,nw));
    coef(n,:)=nc(1,n)*cmn(1,:)+nc(2,n)*cmn(2,:)+nc(3,n)*cmn(3,:);
end
for nn=2:length(wp)
    n=wp(nn);
    coef(n,ea(n))=1;coef(n,we(n))=-1;coef(n,n)=1;
end

%determine result matrix
rm=(-nc(1,:)*vinf(1)-nc(2,:)*vinf(2)-nc(3,:)*vinf(3))';
rm(wp)=0;coef(end+1,bp)=1;rm(end+1)=0; %prevents singular matrix - sum of panel strengths on closed b
ga=coef\rm;

%Determine velocity and pressure at control points
ga=repmat(ga',[3 1]);
for n=1:npanels     %Determine velocity at each c.p. without principal value
    cmn=ffil(rc(:,n),r(:,nw),r(:,sw))+ffil(rc(:,n),r(:,sw),r(:,se))+ffil(rc(:,n),r(:,se),r(:,ne))+ffi                ,nw));
    v(:,n)=vinf+sum(ga.*cmn,2);
end                 %Determine principle value of velocity at each c.p., -grad(ga)/2
gg=v_cross((rc(:,we)-rc(:,no)).*(ga(:,we)+ga(:,no))+(rc(:,so)-rc(:,we)).*(ga(:,so)+ga(:,we))+(rc(:,ea                )+ga(:
te=find([1:npanels]'==ea(:));gg(:,te)=gg(:,te)/2;te=find([1:npanels]'==we(:));gg(:,te)=gg(:,te)/2;

v=v-gg/2; %velocity vector
cp=1-sum(v.^2)/(vinf'*vinf); %pressure
```

Specify free stream and geometry

Compute influence coefficients

Compute result matrix and solve

Compute surface velocity at c.p.s, add principal value

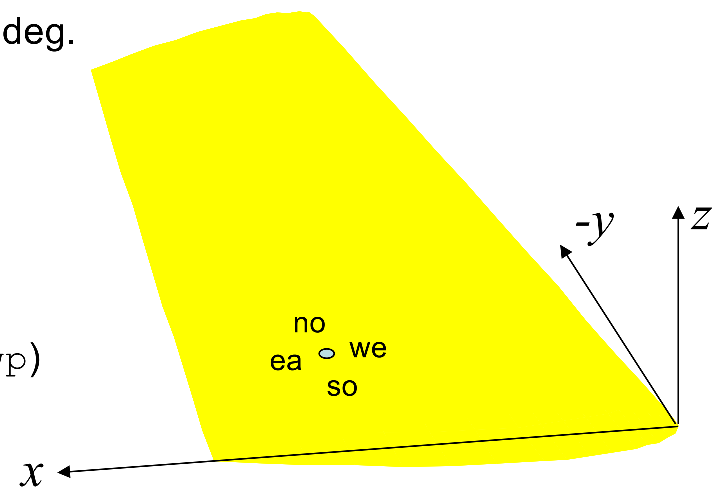
Different Geometry

```
%Specify wing geometry (NACA 0012 section)
xp=[1   0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.25    0.2 0.15 ...
zp=[0   0.01448 0.02623 0.03664 0.04563 0.05294 ...
nb=25;b=2.5;sweep=30;dihedral=15;twist=10;taper=.5;
[r,rc,nw,sw,se,ne,no,we,so,ea,wp,bp]=wing(xp,zp,b,nb,vinf,sweep,dihedral,twist,taper)
xl=-1;xh=2;yl=-1.5;yh=1.5;zl=-1;zh=1;cl=-2;ch=1; % plotting limits
```

• Airfoil coordinates, starting at the trailing and progressing along the upper surface, the leading is at `xp=0`, trailing edge at `xp=1`. Repeat t.e. point at end.

• `nb` is number of points to put across the span
• `b` is span (compared to root chord)
• `sweep`, `dihedral`, `twist` (at tip) are in deg.
• `taper` is taper ratio

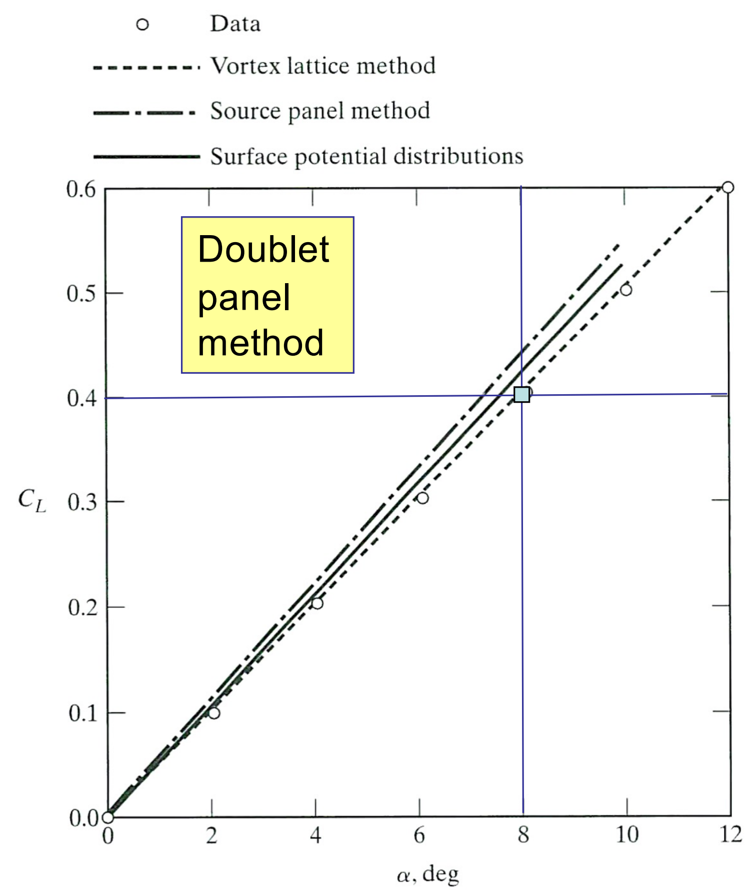
• wing() generates shape
• needs `vinf`, to position wake panels,
• gives panel and control point position vectors (`r` and `rc`) and indices (`nw`, `sw`…) also gives indices of panels in the wake (`wp`) and on the body of the wing (`bp`)



Presenting Forces and Moments

Lift coefficient as a function of angle of attack for a symmetric section, plane wing with a quarter chord sweep of 45 degrees and a taper of 0.5. Margason *et al.* (1985).

Note that the lift curve slope is less that 2π, and decreases with reducing aspect ratio and taper. This is mostly because the tips generate less lift than an equivalent 2D section



Hints on using the code

- Ignore the ‘divide by zero’ and ‘rank deficient’ warnings. These are a consequence of there being a couple of panels (at the t.e. tips) with zero areas, and don’t seem to cause any problems.
- Try to pick numbers of spanwise and chordwise points so aspect ratio of panels doesn’t get extreme - grid independence is most easily checked by increasing or decreasing the number of spanwise points.
- Don’t overkill the number of points (e.g. 200 airfoil points or span points) unless you want to wait to 2025 for results. Remember the solution time goes up as the square of the total number of panels.
- Make sure your airfoil profile specifies leading and trailing edge points.
- Have fun trying out weird configurations, with this and the non-lifting code, and resulting streamline/pressure patterns.