Wings in Ideal Flow
Physics of an Flow Over a Wing

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

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

Bippes, Clark Y, Rectangular Wing 9°, AR=2.4, Re=100000
Physics of Flow Over a Wing

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 shouldn’t balance in the streamwise direction, when that wing is generating lift
In both cases this is equivalent to requiring that the surface vorticity parallel to the t.e. \( \Omega_p \) 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

Taper assumed linear from zero to root, unless otherwise specified.

\[ V_\infty = iu_\infty + jv_\infty + kw_\infty \]
Wing Nomenclature

Angles

\[ \alpha = \arctan \left( \frac{w_\infty}{u_\infty} \right) \]

Also sideslip angle
\[ \beta = \arctan \left( \frac{v_\infty}{u_\infty} \right) \]
(zero as pictured)

Positive twist increases \( \alpha \), and is termed 'wash-in'. Negative twist is termed 'wash-out'. Assumed to linearly increase from zero at root, unless otherwise specified.
Wing Nomenclature

Forces and Moments

\[ V_\infty = \left| \mathbf{V}_\infty \right| = \sqrt{u_\infty^2 + v_\infty^2 + w_\infty^2} \]

\[ C_L = \frac{L}{\frac{1}{2} \rho V_\infty^2 S} \quad C_D = \frac{D}{\frac{1}{2} \rho V_\infty^2 S} \quad C_M = \frac{M}{\frac{1}{2} \rho V_\infty^2 S c} \]

Moment conventions

Us

\[ M_{\text{pitch}} = M \quad C_P = C_M \]

\[ M_{\text{yaw}} = -N \quad C_Y = -C_N \]

\[ M_{\text{roll}} = -L \quad C_R = -C_L \]

Ind. std
Computing Forces and Moments

Planform area

\[ S = \frac{1}{2} \int_{\text{Surface}} |\mathbf{k} \cdot \mathbf{n}| \, dS = \frac{1}{2} \int_{\text{Surface}} |\mathbf{k} \cdot dA| \]

Total Force

\[ \mathbf{F} = F_x \mathbf{i} + F_y \mathbf{j} + F_z \mathbf{k} = \]

as coefficients

\[ C_F = \frac{\mathbf{F}}{\frac{1}{2} \rho U_{\infty}^2 S} = C_x \mathbf{i} + C_y \mathbf{j} + C_z \mathbf{k} = -\frac{1}{S} \int_{\text{Surface}} C_p \, dA \]

Total moment about origin of \( \mathbf{r} \)

\[ \mathbf{M} = M_{\text{roll}} \mathbf{i} + F_{\text{pitch}} \mathbf{j} + F_{\text{yaw}} \mathbf{k} = -\int_{\text{Surface}} \mathbf{p} \times \mathbf{n} \, dS = -\int_{\text{Surface}} \mathbf{p} \times dA \]

as coefficients

\[ C_M = \frac{\mathbf{M}}{\frac{1}{2} \rho U_{\infty}^2 SC} = C_R \mathbf{i} + C_P \mathbf{j} + C_Y \mathbf{k} = -\frac{1}{SC} \int_{\text{Surface}} C_p \mathbf{r} \times dA \]

Measured from \( \frac{1}{4} \) chord at root, so moments about this point

\( C_L \) and \( C_D \) obtained by rotating \( C_z \) and \( C_x \) by the angle of attack
Ideal Flow Methods for Wings

- Panel Methods
- Lifting Surface Methods
- Vortex Lattice Method
- Lifting line theory

See Katz and Plotkin for complete descriptions and prescriptions
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)
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.

Ideally wake panels extend to $\infty$. In practice a big distance (50c?) is enough.
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 0.2 0.1 0.075 0.05 0.025 0.0125 0]
zp=[0 0.01448 0.02623 0.03664 0.04563 0.05294 0.05803 0.06002 0.05941 0.05737 0.05345 0.0468 0]

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 counterclockwise)
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(:,i));
    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 <
gac=coef\rm;

%Determine velocity and pressure at control points
ga=repmat(gac',[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(:,i));
        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)))+
    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

%Extend influence coefficient
matrix to include strength relation for each wake panel
Different geometry
Compute influence coefficient only for panels on the body of the wing
Fix to improve ∇Γ calculation near t.e.
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)
Under the hood

**Function**
```
[r, rc, nw, sw, se, ne, no, we, so, ea, wp, bp] = wing(xp, zp, b, nb, vinf, sweep, dihedral, twist, taper)
```

```
sweep = sweep * pi / 180; dihedral = dihedral * pi / 180; twist = twist * pi / 180;
```

% Define vertices of panels
```
yp = ([0:nb+1]-1) / (nb-1) * b - b/2; % evenly spaced points spanwise, extra points used to close tips
x = repmat(xp, [nb+2 1]) - 0.25; % grid of x points, shift quarter chord to origin
y = repmat(yp', [1 length(xp)]);
z = repmat(zp, [nb+2 1]);
z(1,:) = 0; z(end,:) = 0; y(1,:) = y(2,:); y(end,:) = y(end-1,:); % close tips
```

% Apply linear taper
```
z = z.*(1+2*(taper-1)/b*abs(y));
x = x.*(1+2*(taper-1)/b*abs(y));
```

% Apply linear twist
```
lt = (twist*2*abs(y)/b);
z1 = z.*cos(lt) - x.*sin(lt);
x1 = x.*cos(lt) + z.*sin(lt);
x = x1; z = z1;
```

% Apply dihedral
```
z = z + abs(y) * tan(dihedral);
x = x + abs(y) * tan(sweep);
```

% Return leading edge to origin
```
x = x + .25;
```

% Add wake panels for trailing edge Kutta condition. Wake panels end up as last array column
```
xwakelength = 50; % number of chordlengths to extend wake panels downstream of t.e.
mvinf = sqrt(vinf(1)^2 + vinf(2)^2);
x(:,end+1) = x(:,end) + vinf(1)/mvinf * wakelength; % Extend wake in the plane of the root chord
y(:,end+1) = y(:,end) + vinf(2)/mvinf * wakelength; % (but with any side slip of the free stream)
z(:,end+1) = z(:,end);
```

...all that indexing stuff down here

```
makes flat untwisted rectangular wing of correct span, origin at 1/4 chord
```

```
adds wake panels
```
% 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(:,se),r(:,sw),r(:,we),r(:,ne))+
    cmn(1,:)+cmn(2,:);
    coef(n,:)=nc(1,n)*cmn(1,:)+nc(2,n)*cmn(2,:);
end
for nn=2:length(wp)
    n=wp(nn);
    coef(n,ea(n))=1;coef(n,we(n))=-1;coef(n,n)=1;
end

For wake panels the east and west indices are used to identify the two trailing edge panels.

\[ \Gamma_W = \Gamma_L - \Gamma_U \]
Computing Forces and Moments

```matlab
% Compute forces and moments
cp1 = repmat(cp,[3 1]); area = sum(abs(ac(3,bp)))/2; cbar = area/b; alpha = atan(vinf(3)/vinf(1));
rc1 = rc; rc1(1,:) = rc1(1,:) - 0.25; % Compute moments about quarter chord
Cf = sum(-cp1(:,bp).*(ac(:,bp)),2)/area; % Forces are the -\text{Integral}(\text{pressure} \ d(\text{area vector}))
Cm = sum(-cp1(:,bp).*v_cross(rc1(:,bp),ac(:,bp)),2)/area/cbar; % Moments are the -\text{Integral}(\text{pressure position} \ x \ d(\text{area vector}))
Cl = Cf(3)*cos(alpha) - Cf(1)*sin(alpha); Cd = Cf(3)*sin(alpha) + Cf(1)*cos(alpha);
```
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 than $2\pi$, and decreases with reducing aspect ratio. This is mostly because the tips generate less lift than an equivalent 2D section.
Presenting Forces and Moments
Presenting Forces and Moments

Computed drag not very accurate – this is a well documented issue when integrated pressures are used to determine induced drag. More accurate estimates of $C_D$ are usually made by ....
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 errors, 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 2009 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.