JavaFoil Theory Document

Contents

Panel Method

JavaFoil implements a classical panel method with linearly varying vorticity distribution. This is somewhere between the panel methods in *XFOIL* (constant vorticity per panel) and Epplers *PROFIL* code (parabolic variation of vorticity). The resulting equation system consists therefore of a (# of panels +1)² sized matrix and two right hand sides. These are for 0° and 90° angle of attack and can be solved for the two corresponding vorticity distributions efficiently. The vorticity distribution for any arbitrary angle of attack is then derived from these two solutions (remember that potential theory is linear and allows for superposition). There is no interaction with the boundary layer, as in XFOIL, though.

$$\begin{bmatrix} C_{1,1} & \cdots & C_{N+1,1} \\ \vdots & \ddots & \vdots \\ C_{1,N} & \cdots & C_{N+1,N} \\ 1 & \cdots 0 \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} \gamma_{1,0°} & \gamma_{1,90°} \\ \gamma_{2,0°} & \gamma_{2,90°} \\ \vdots & \vdots \\ \gamma_{N+1,0°} & \gamma_{N+1,90°} \end{bmatrix} = \begin{bmatrix} RHS_{1,0°} & RHS_{1,90°} \\ RHS_{2,0°} & RHS_{2,90°} \\ \vdots & \vdots \\ RHS_{N+1,0°} & RHS_{N+1,90°} \end{bmatrix}$$

Critical Pressure Coefficient

When the local pressure somewhere on the airfoil surface drops below the critical pressure, the flow speed exceeds the speed of sound.
When supersonic speed is exceeded anywhere on the surface, the character of the flow may change dramatically. In most cases pressure recovery from supersonic to subsonic velocities (from $C_p < C_{p,crit}$ to $C_p > C_{p,\,crit}$) is to leading abrupt recompression with shocks. The analysis of these flow fields requires more complex methods, capable of handling compressible flows (e.g. by solving the full, compressible potential equations or by solving the Euler equations).

In *JavaFoil*, the critical pressure coefficient is calculated from the relation

$$C_{p,\,crit} = \frac{2}{\kappa \cdot M_\infty^2} \cdot \left(\left(\frac{2}{\kappa + 1} \cdot \left(1 + \frac{\kappa - 1}{2} \cdot M_\infty^2 \right) \right)^{\frac{\kappa}{\kappa - 1}} - 1 \right)$$

(Küchemann, „The Aerodynamic Design of Aircraft“, p.115).

In terms of the velocity ratio the critical limit is found from

$$\left(\frac{v}{v_\infty}\right)_{crit} = \sqrt{1+2\cdot\frac{1+M_\infty^2}{(\kappa+1)\cdot M_\infty^2}}$$

(Küchemann, „The Aerodynamic Design of Aircraft“, p.114).

Compressibility Corrections

There are different ways to correct incompressible flow results for compressibility effects. One should keep in mind that these are only corrections – they can never produce the correct physical effects. Therefore the applicability of all compressibility corrections is limited to cases where the local flow velocity (which can be much higher than the onset flow velocity) is well beyond the speed of sound. In practical application one can use such corrections up to maybe Mach = 0.6, but the error grows rapidly when Mach exceeds 0.8.

In *JavaFoil*, the panel analysis is always running on the given airfoil shape – it is never geometrically distorted. Compressibility corrections are applied later to the local surface velocities according to the Kàrmàn-Tsien approximation (usually written for C_p)

$$\left(\frac{v}{v_\infty}\right)_{comp.} = \left(\frac{v}{v_\infty}\right)_{inc.} \cdot \frac{1-\frac{M_\infty^2}{2-M_\infty^2}}{1-\frac{M_\infty^2}{2-M_\infty^2}\cdot\left(\frac{v}{v_\infty}\right)_{inc.}^2} \; .$$

(Cebeci, "An Engineering Approach to the Calculation of Aerodynamic Flows", p. 32).

Boundary Layer Analysis

The boundary layer analysis module implements an integral boundary layer integration scheme following publications by Prof. R. Eppler.

Note: the local skin friction coefficient as given on the Boundary Layer card is twice the value as used by Eppler to follow the more common convention $C_f = \tau_0 / \left(\frac{\rho}{2}\cdot v_\infty^2\right)$.

In *JavaFoil* there is no interaction between the boundary layer and the external flow, as in *XFOIL*, though. Therefore largely separated flows cannot be analyzed – a short flow separation (s_{sep} < 10%) at the trailing edge does not affect the results very much. Also laminar separation bubbles are not modeled; when laminar separation is detected the code switches to turbulent flow.

Transition Criteria

Methods to predict transition from laminar to turbulent flow have been developed by many authors since the early days of Prandtl’s boundary layer theory. While it is possible to analyze the stability of a boundary layer numerically, all methods which are practical and fast are more or less approximate and rely on empirical relations (usually derived from experiments).

Methods based on $n - Re_{\delta_2}$ envelopes

Drela approximates the envelopes of the amplification rate n versus Re_{δ_2} by straight lines of the form $\tilde{n} = f\left(\mathrm{Re}_{\delta_2}, H_{12}\right)$. Two versions of this approximation were used in his codes of the *XFOIL* and *MSES/ISES* family.

The approximation is expressed by

$$\tilde{n} = \frac{\partial \tilde{n}}{\partial Re_{\delta_2}} \cdot \left(Re_{\delta_2} - Re_{\delta_2,crit}\right).$$

Transition can occur when $Re_{\delta_2} > Re_{\delta_2,crit}$ and $\tilde{n} > n_{crit}$.

XFOIL 1.1 and 5.4

$$\frac{\partial \tilde{n}}{\partial Re_{\delta_2}} = 0.01 \cdot \sqrt{\left(2.4 \cdot H_{12} + 2.5 \cdot \tanh\left(1.5 \cdot H_{12} - 4.65\right) - 3.7\right)^2 + 0.25}$$

$$\log_{10}\left(Re_{\delta_2,crit}\right) = \left(\frac{1.415}{H_{12} - 1} - 0.489\right) \cdot \tanh\left(\frac{20}{H_{12} - 1} - 12.9\right) + \frac{3.295}{H_{12} - 1} + 0.44$$

[described in Mark Drela, Giles, AIAA-86-1786-CP, 1986
also cited in Xiao-liang Wang, Xue-xiong Shan, “Shape Optimization of Stratosphere Airship”, Journal of Aircraft V43N1, 2006.]

XFOIL 5.7

Modification in 1991

$$\frac{\partial \tilde{n}}{\partial Re_{\delta_2}} = 0.028 \cdot \left(H_{12} - 1\right) - \frac{0.0345}{e^{-\left(\frac{3.87}{H_{12}-1} - 2.52\right)^2}}$$

$$\log_{10}\left(Re_{\delta_2,crit}\right) = 0.7 \cdot \tanh\left(\frac{14}{H_{12} - 1} - 9.24\right) + 2.492 \cdot \left(\frac{1}{H_{12} - 1}\right)^{0.43} + 0.66$$

XFOIL 6.8

only a tiny modification (term 0.66 -> 0.62)

$$\frac{\partial \tilde{n}}{\partial Re_{\delta_2}} = 0.028 \cdot \left(H_{12} - 1\right) - \frac{0.0345}{e^{-\left(\frac{3.87}{H_{12}-1} - 2.52\right)^2}}$$

$$\log_{10}\left(Re_{\delta_2,crit}\right) = 0.7 \cdot \tanh\left(\frac{14}{H_{12} - 1} - 9.24\right) + 2.492 \cdot \left(\frac{1}{H_{12} - 1}\right)^{0.43} + 0.62$$

Method of Arnal:

A set of tables produced by D. Arnal has been approximated by W. Würz with polynomials:

$$\frac{\partial \tilde{n}}{\partial Re_{\delta_2}} = a_1 + a_2 \cdot H_{12} + a_3 \cdot H_{12}^2$$

$$\log_{10}\left(Re_{\delta_2,crit}\right) = b_1 + b_2 \cdot H_{12} + b_3 \cdot H_{12}^2$$

(see doctoral thesis of Werner Würz, IAG Stuttgart)

Abbreviations:

approximation of n	$\tilde{n}$
displacement thickness	δ_1
momentum thickness	$\delta_2 = \theta$
shape factor displacement thickness / momentum thickness	$H_{12} = \frac{\delta_1}{\delta_2}$
Reynolds number based on local momentum thickness	$\mathrm{Re}_{\delta_2} = \mathrm{Re}_\theta$

Stall Corrections

Emprirical Stall Correction #1 („CalcFoil“)

if ($\alpha > 0$)

{

// handle separation on upper surface

// drag increment

$$C_{d,upper} = C_{d,upper} + \left|\sin^2\alpha \cdot \left(x_{TE} - x_{sep,upper}\right)^2 + 0.025 \cdot \cos\alpha \cdot \left(x_{TE} - x_{sep,upper}\right)^2\right|$$

// lift multiplier reduces lift linearly with length of separated length

$$C_\ell = C_\ell \cdot \left(1 - 0.2 \cdot \left(x_{TE} - x_{sep,upper}\right)\right)$$

}

else if ($\alpha < 0$)

{

// handle separation on lower surface

// drag increment

$$C_{d,lower} = C_{d,lower} + \left|\sin^2\alpha \cdot \left(x_{TE} - x_{sep,lower}\right)^2 + 0.025 \cdot \cos\alpha \cdot \left(x_{TE} - x_{sep,lower}\right)^2\right|$$

// lift multiplier reduces lift linearly with length of separated length

$$C_\ell = C_\ell \cdot \left(1 - 0.2 \cdot \left(x_{TE} - x_{sep,lower}\right)\right)$$

}

// moment multiplier

$$C_{m,corrected} = C_{m,\ panel\ method} \cdot 0.9 \cdot x^2_{sep,lower} \cdot x^2_{sep,upper}$$

// lift multiplier due to suction peak criterion

$$C_\ell = C_\ell \cdot \frac{1}{\left(\frac{\Delta C_{P,\max}}{20}\right)^2 + 1}$$

Emprirical Stall Correction #2 („Eppler“)

if ($\alpha > 0$)
{
// handle separation on upper surface
if ($x_{sep,upper} < x_{TE}$)
{
// trailing edge angle of upper surface

$$\theta_{TE} = \arctan\left(-\frac{y_{sep,upper} - y_{TE}}{x_{sep,upper} - x_{TE}} \right)$$

}
else
{

$$\theta_{TE} = 0$$

}
// drag increment

$$C_{d,upper} = C_{d,upper} + 0.2 \cdot \sin\left(\alpha + \theta_{TE}\right) \cdot \left(x_{TE} - x_{sep,upper}\right)^2$$

$$\Delta C_\ell = C_{l,\max,\ fudge} \cdot \left(\alpha + \theta_{TE}\right) \cdot \pi \cdot \left(x_{TE} - x_{sep,upper}\right)$$

if ($\Delta C_\ell > 0$)
{
// lift reduction

$$C_\ell = C_\ell - \Delta C_\ell$$

}
else
{
// lift multiplier

$$C_\ell = C_\ell \cdot \left(1 - \sin\alpha \cdot \left(x_{TE} - x_{sep,upper}\right)\right)$$

}

// moment increment

$$C_m = C_m - \sin\alpha \cdot \left(x_{TE} - x_{sep,upper}\right) \cdot \left(0.5 \cdot \left(1 + x_{sep,upper}\right) - 0.25\right)$$

}
else if ($\alpha < 0$)
{
// handle separation on lower surface
if ($x_{sep,lower} < x_{TE}$)
{
// trailing edge angle of lower surface

$$\theta_{TE} = \arctan\left(-\frac{y_{sep,lower} - y_{TE}}{x_{sep,lower} - x_{TE}} \right)$$

}
else
{

$$\theta_{TE} = 0$$

}

// drag increment

$$C_{d,lower} = C_{d,lower} - 0.2 \cdot \sin\left(\alpha + \theta_{TE}\right) \cdot \left(x_{TE} - x_{sep,lower}\right)^2$$

$$\Delta C_\ell = C_{l,\max,\,fudge} \cdot \left(\alpha + \theta_{TE}\right) \cdot \pi \cdot \left(x_{TE} - x_{sep,lower}\right)$$

if ($\Delta C_\ell < 0$)

{

// lift reduction

$$C_\ell = C_\ell - \Delta C_\ell$$

}

else

{

// lift multiplier

$$C_\ell = C_\ell \cdot \left(1 - \sin\alpha \cdot \left(x_{TE} - x_{sep,lower}\right)\right)$$

}

// moment increment

$$C_m = C_m - \sin\alpha \cdot \left(x_{TE} - x_{sep,lower}\right) \cdot \left(0.5 \cdot \left(1 + x_{sep,lower}\right) - 0.25\right)$$

}

// lift multiplier due to modified suction peak criterion

$$C_\ell = C_\ell \cdot \frac{1}{\left(\frac{\Delta C_{P,\max}}{30}\right)^2 + 1}$$

Finite Wings in *JavaFoil*

JavaFoil is a program for the analysis of two dimensional airfoils. Nevertheless it supports a very simple approximate model of finite wings. When the user supplies a value for the aspect ratio on the Options card simple formulas are used to determine an approximation of the 3D effects.

These effects are applied to the polars produced by *JavaFoil* and make it possible to get a first impression of the relations between induced drag and airfoil drag. For example the importance of the airfoil drag is diminishing for higher lift coefficients.

No additional wing effects (like Reynolds number variation due to taper) are taken into account. For more detailed studies a 3D wing analysis code must be used (e.g. for subsonic flow lifting line, vortex lattice or panel methods).

The three dimensional corrections can be applied to the results for constant Reynolds number (Polar card) as well as more realistically for the results associated with a constant wing loading (Aircraft card).

Polars for a constant Wing Loading

Airfoil data has traditionally been presented in form of graphs and tables for constant Reynolds numbers. This form results from the typical way wind tunnel experiments and numerical analyses are conducted. In a wind tunnel it is relatively easy to maintain a constant wind speed and Reynolds number.
Now the lift coefficient of a real airplane depends on the speed because the wing loading is usually fixed during flight – flying at low lift coefficients results in high speeds (and high Reynolds numbers) and vice versa. Therefore the operating points during flight would slice through a set of polars having constant Reynolds numbers.
It is possible to create polars more closely related to the conditions during flight. This would require adjusting the wind speed to each lift coefficient, which is cumbersome and expensive in a wind tunnel, but feasible in a numerical tool like *JavaFoil*.

Abbreviations:

mass of aircraft	m	kg
gravity constant	g	m/s^2
density of medium	ρ_∞	m/s^2
kinematic viscosity	υ	m^2/s
flight speed	v_∞	m/s
wing area	S	m^2
chord length	ℓ	m
Reynolds number	Re	-

Basic Equations

The definition of the lift coefficient is $C_L = \frac{m \cdot g}{\frac{\rho_\infty}{2} \cdot v_\infty^2 \cdot S}$. Solving the definition of the Reynolds number $\mathrm{Re} = \frac{v_\infty \cdot \ell}{\nu}$ for the velocity v_∞ yields $v_\infty = \frac{\mathrm{Re} \cdot \nu}{\ell}$. Inserting this result into the definition of the lift coefficient produces

$$C_L = \frac{m \cdot g \cdot \ell^2}{\frac{\rho_\infty}{2} \cdot \mathrm{Re}^2 \cdot \nu^2 \cdot S} \quad .$$

Solving for the Reynolds number yields

$$\mathrm{Re} = \sqrt{\frac{g}{\frac{\rho_\infty}{2} \cdot C_L \cdot \nu^2} \cdot \frac{m}{S} \cdot \ell^2} \text{ or } \mathrm{Re} = \frac{\ell}{\nu} \cdot \sqrt{\frac{2 \cdot g}{\rho_\infty \cdot C_L} \cdot \frac{m}{S}} \quad .$$

Using these results an aircraft oriented airfoil polar for a given wing loading $\frac{m}{S}$ and given mean chord length ℓ can be derived. Due to the dependency between lift coefficient and Reynolds number an iteration scheme must be used. The calculation procedure for a polar would be as follows:

- prescribe the environmental condition density ρ_∞ and kinematic viscosity ν.
- prescribe a wing loading $\frac{m}{S}$ and a reference chord length ℓ.

- perform the following calculation sequence:

inital value

$$Re^* = 10^6$$

$$for\ (\alpha = \alpha_0\ to\ \alpha_1\ step\ \Delta\alpha)$$

{

iterate

{

$$Re = Re^*$$

$$C_L = f(\alpha, Re)$$

$$Re^* = \sqrt{\frac{g}{\frac{\rho_\infty}{2} \cdot C_L \cdot v^2} \cdot \frac{m}{S} \cdot \ell^2}$$

}

$$while\left(\frac{\left|Re^* - Re\right|}{Re} > \varepsilon\right)$$

}

A precaution must be undertaken to handle cases where $C_L \to 0$. Here JavaFoil limits the Reynolds number to a value corresponding to a small lift coefficient, e.g. $C_L = 0.02$.

Note: One can also derive the Reynolds number for a constant ratio $\frac{m}{\Lambda}$, eliminating the chord length ℓ. This has not been implemented in *JavaFoil* as it was considered more abstract to think in terms of $\frac{m}{\Lambda}$ instead of the aircraft design parameters $\frac{m}{S}$ and ℓ. But as the relation is $\frac{m}{S} \cdot \ell^2 = \frac{m}{\Lambda}$ it would be sufficient to use $\frac{m}{\Lambda}$ instead of $\frac{m}{S}$ in *JavaFoil* while setting $\ell = 1$.

Correction of Lift for given Aspect Ratio and Mach number

For a given angle of attack, a 3D wing of finite aspect ratio produces less lift than the 2D airfoil section, which corresponds to an infinite aspect ratio. Another correction has to be applied when the Mach number is larger than zero. In subsonic flight more lift is produced when the Mach numbers is increased.

The following correction is applied to the lift coefficient of a 2D airfoil C_ℓ in order to approximate the lift coefficient C_L of a 3D wing in compressible flow. The correction is divided into two regimes of aspect ratios.

For small aspect ratios ($\Lambda < 4$) the following formula is used:

$$C_L = \frac{C_\ell}{\sqrt{1 - M_\infty^2 + \left(\frac{2 \cdot \pi}{\Lambda \cdot \pi}\right)^2} + \frac{2 \cdot \pi}{\Lambda \cdot \pi}}$$

If the aspect ratio is larger, $\Lambda \geq 4$, the simplified approximation is applied:

$$C_L = \frac{C_\ell}{\sqrt{1-M_\infty^2} + \frac{2\cdot\pi}{\Lambda\cdot\pi}}$$

Implementation in *JavaFoil*

```
public final static double LiftForAspectRatio(double dCl,
                                              double dAspectRatio,
                                              double dMachNumber)
    {
        double dReturn = dCl;

        // correction for finite wings
        if (dAspectRatio > 0.1)
        {
            // Source: Anderson, "Aircraft Performance and Design"
            // lift gradient reduction factor
            // a 0 / (pi*AR)
            double dGradientRatio = 2.0 * Math.PI / (Math.PI *
                                         dAspectRatio);
            if (dAspectRatio < 4.0)
            {
                // low aspect ratio, compressible (Anderson [2.18b])
                dReturn /=
                         (Math.sqrt(1.0 - Math.pow(dMachNumber, 2.0) +
                                    Math.pow(dGradientRatio, 2.0)) +
                          dGradientRatio);
            }
            else
            {
                // high aspect ratio, compressible (Anderson [2.16])
                dReturn /=
                         (Math.sqrt(1.0 - Math.pow(dMachNumber, 2.0)) +
                          dGradientRatio);
            }
        }
        return (dReturn);
    }
```

Determination of Drag for given Aspect Ratio and Mach number

After the lift coefficient of the 2D airfoil for a given angle of attack α has been corrected to the properties of the 3D wing, an approximation of the induced drag is added to the airfoil drag (for the same angle of attack α). As no information about the real wing shape is available, the assumption of having a "good" wing planform is assumed.
Therefore the induced drag component is calculated by using the classical formula derived by lifting line theory (Prandtl).

$$C_{D,i} = k\cdot\frac{C_L^2}{\pi\cdot\Lambda}$$

In *JavaFoil* the "k-Factor" is assumed to be 1.0 (planar wing with elliptical lift distribution).

Implementation in *JavaFoil*

```
    public final static double DragForAspectRatio(double dCd, double dCl,
                                                  double dAspectRatio,
                                                  double dMachNumber)
    {
```

```
        double dReturn = dCd;

        if (dAspectRatio > 0.1)
        {
            // add the induced drag of finite wing according to Prandtl
            dReturn += dCl * dCl / (Math.PI * dAspectRatio);
        }

        return (dReturn);
    }
```