스플라인 및 **NURB** 곡선을 통해 재생성한 **Discretized Morphing** 에어포일의 공력특성에 관한 연구

A Study on the Aerodynamic Characteristics of Discretized Morphing Airfoils Regenerated by Splines and NURB Curves

2003년 10월 일

목 차

List of Figures

Abstract

A Study on the Aerodynamic Characteristics of Discretized Morphing Airfoils Regenerated by Splines and NURB Curves

The study focuses on the generation of airfoil panels, the reconfiguration of the morphing airfoils and its method. The first assumption is the air and its flow are assumed incompressible, irrotational, inviscid and subsonic for simplicity. The second assumption made is the morphing airfoils form the airfoil shapes generated by conventionally existing low thickness/chord(T/C) ratio and high T/C ratio airfoils. The next assumption made is, thereon, since airfoils in continuous morphing stages are generated from both proper airfoils, they should, therefore, show proper pressure distributions; that is the regenerated morphing airfoils are apt for flying objects as UCAVs and general airplanes. For analysis, the continuous morphing stages are discretized.

To regenerate the airfoil, numerical curve fitting methods such as natural cubic spline, piecewise polynomial fitting, and NURB curves were tested successively in a routine for better pressure distribution, and for higher efficiency. In the course of panel regeneration, airfoils by NURB showed positive tendency that it generated dense panels at both leading and trailing edges while maintaining proper Cp distribution, leaving wide panels at the middle without further vigorous manipulation. The pressure distribution graphs using the linear strength vortex panel(LSVP) method were obtained as the results. In order to verify the results, XFOIL was used for validification. The study finds the concept of morphing airfoils and its approaching method described in this study feasible. The study also suggests another method for finding 'aerodynamic center' by deriving the positional (total) moment coefficient ($c_{m,@x_n}$) on the chord.

Key Word: Morphing, Airfoil, Spline, NURB, Regeneration, Aerodynamic Center, Linear Strength Vortex Panel Method

1. 서론(Introduction)

1.1. 연구배경

현존하는 비행체의 날개는 플랩으로 작동되는 고정형 비행체가 주종을 이루고 있으며 에어포일의 단면은 해당 비행체가 주로 머물고 있는 속도대에 최적화되어 있다. 이는 다음에 제시되는 fig.1에서 확인할 수 있다. 그러므로 현 비행체의 에어포일은 고속 비행체의 경우 저속에서의 효율성이 고속에서 운용되는 경우보다 상대적으로 낮을 것으로 보이며, 그 반대의 경우도 마찬가지이다. 그러나 항공기 이착륙에는 저속에서 고속까지 혹은 고속에서 저속까지의 속도 변화를 필수적으로 수반하게 되어있으며, 이상적인 미래의 중형 UCAV와 고속 비행체는 극단적인 비행 형태를 요구할 것이다. 민간 수송기인 경우에는 고양력 장치인 다단계 플랩으로 이를 구현하고 있으며, 이를 위해 수반되어야 하는 부수적인 기계 장치와 동력에 의해서 효율성이 상대적으로 낮을 것으로 보인다. 만일 에어포일을 복잡한 기계적인 장치를 수반하지 않고 공압이나 엑츄에이터로 조절하여 양력을 높이는 압력계수를 불러올 수 있다면 속도 및 받음각에 대한 비행 효율은 일반 고정 날개보다 더 개선될 수 있을 것이며, 이러한 구조에 있어서 완전히 새로운 디자인이 아닌 약간의 날개 구조 변화만 필요하다면 효율성 향상도 기대해 볼 만 하다. 또한 에어포일 단면을 조정함으로써 높은 받음각에서도 매끈한 흐름이 가능해질 수 있다(Ref.[4], p.332). 이를 증명하듯이 NASA에서는 최근에 모핑 기법을 미래 기술 중 하나로 꼽았다. Morphing wing이 가져올 수 있는 효과는 최소의 변형으로 해당 속도에서 planform 넓이를 조절함으로서 항력과 양력을 최적화할 수 있고, 이착륙시에는 단면적 변화로 최대 양력에 해당하는 받음각을 연장시킬 수 있으며, 부수적으로 양날개 단면의 압력차를 이용하여 롤링 효과를 얻을 수 있다. 새로운 디자인이 아닌 공압과 엑츄에이터로 구현할 수 있다면 기존 날개에 비해 더 향상된 효율을 보일 것으로 예측된다.

1.2. 최근 연구동향

최근 연구동향은 방법론, 비행 타당성, 효율성 및 최적화를 위주로 동시에 진행되고 있다. Prock[1]은 에어포일 모핑에 필요한 최적의 액츄에이터 에너지에 대해 조사하였다. Gano와 Renaud[2]는 variform wing concept를 이용하여 더 높은 연료 효율을 얻는 방법에 대해 기술하였으며, Cadogan[3]은 날개에 대해 공압을 이용해 확장과 변형이 용이하게 하는 연구를 수행하였다.

1.3. 동기

밑에 그림에서 본 바와 같이 각 비행체들은 최적의 속도대가 있으며 에어포일도 이에 상응한 형태를 가지고 있다. 가장 기본적인 질문은, AR비와 형태가 고정되더라도 저속 에어포일에서 고속 에어포일로 혹은 그 반대로 단면을 변형시킬 수 있는지 여부이다.

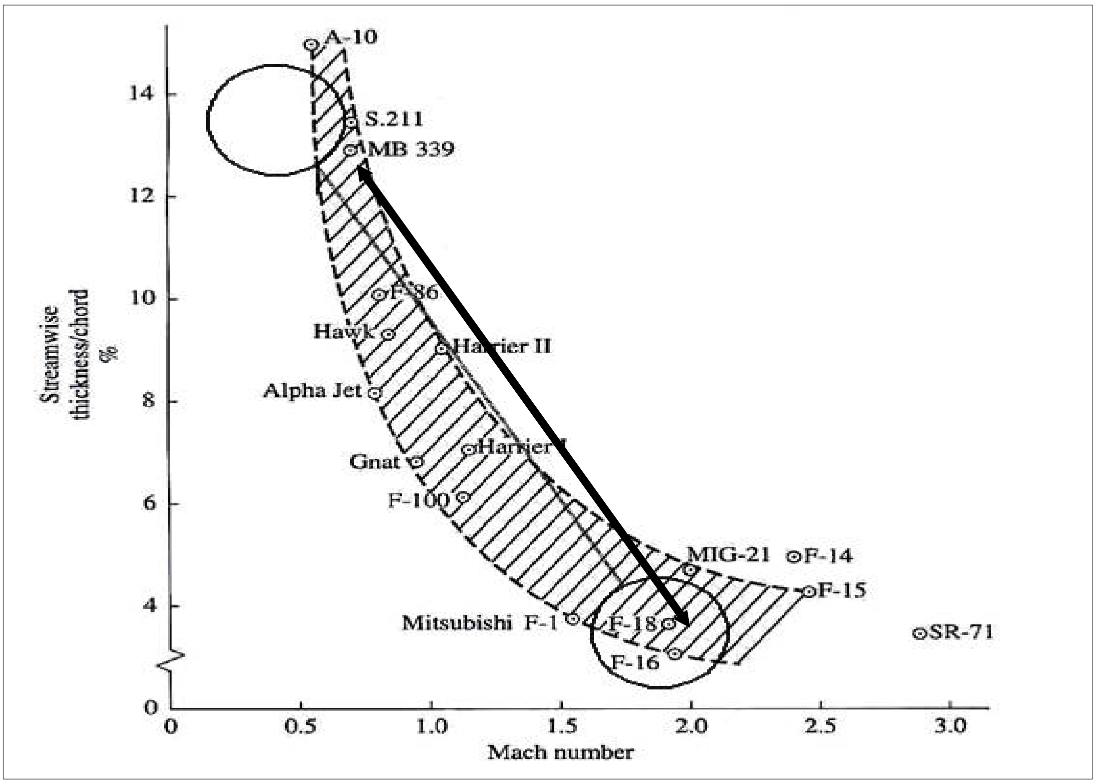

Fig.1 Graph showing thickness/chord ratio versus speed(Ref.[4])

1.4. 목표

에어포일 모핑의 타당성 조사와 접근 방법을 제시하는 것이 주요 목표이다. 먼저 가정으로는 기체 흐름은 비회전, 비압축, 비점성의 가장 이상적인 환경이며 아음속 영역내로 제한한다. 다음 가정은 두 에어포일에 의해 생성된 Morphing단계의 에어포일은 비행에 적합한 압력 분포, 즉 매끈한 흐름을 보인다고 가정하며, 연속적인 Morphing 단계는 분석을 위해 이산화 시킨다. 프로그램이 과정의 일부분으로 수행하도록 한다. 생성된 에어포일에 대한 압력 분석이 필요하며 이는 기존에 있는 수치해석 방법(Linear Strength Vortex Panel Method)을 이용하며 XFOIL로 검증하기로 한다. 마지막으로 압력중심과 공력중심에 대해 간단히 조사한다.

구체적으로, 기준 에어포일로 고속 전투기(F-16)에 사용되는 NACA64a204에 대한 압력분포를 조사한다. 에어포일 하반부분이 기하학적으로 비슷한 고 thickness/ chord(T/C) 비의 에어포일을 선정한다. 고 T/C비 에어포일 상반부분에 대해 곡선을 생성한 후, 기준 에어포일과 선정된 에어포일의 상단 차이를 임의 개수의 선으로 균등하게 분할하여 연속적인 Morphing 단계를 이산화 Morphing 단계로 변형 단계를 추출한다. 물론 에어포일 구조상 상반 에어포일의 앞전이 회전이동을 하여 이에 대한 압력 분포를 조사하여야 하나 매끄러운 유동을 형성하여야 하기 때문에, 결국 이론적으로 흐름이 매끈하지 않은 에어포일은 공압으로 보상되어 위에 기술된 Morphing 단계의 매끈할 것으로 가정한 Morphing 에어포일을 형성해야한다. 해당하는 새로운 Morphing단계의 에어포일을 구한 후 받음각에 대한 각 이산 에어포일의 압력 분포 및 공력계수, 압력중심과 공력중심에 대해 고찰한다. 단, 항력, 공탄성 및 기타 계수는 계산의 단순성을 위해 제외한다.

2. 기본 이론(Basic Theory)

2.1. 비회전, 비압축성 유동에 대한 지배방정식

비압축성 유동에 대한 속도의 조건은 연속방정식에서 $\nabla \cdot V = 0$와 같다. 비회전 비압축 유동에 대한 지배방정식은 라플라스 방정식이며 우선 이를 위해 속도 포텐셜을 다음과 같이 정의한다. $V = \nabla \Phi^{*}$. 이를 비압축성 조건에 대입하면,

$$\nabla \cdot (\nabla \Phi^{*}) = 0 \qquad [2.1]$$

$$\nabla^{2} \Phi^{*} = 0 \qquad [2.2]$$

직교좌표계에서는 다음과 같다.

$$\nabla^{2} \Phi^{*} = \frac{\partial^{2} \Phi^{*}}{\partial x^{2}} + \frac{\partial^{2} \Phi^{*}}{\partial y^{2}} + \frac{\partial^{2} \Phi^{*}}{\partial z^{2}} = 0 \qquad [2.3]$$

또한 유동함수 Ψ는 다음과 같이 나타낼 수 있다.

$$u = \frac{\partial \Psi}{\partial z}, \quad v = -\frac{\partial \Psi}{\partial x} \qquad [2.4]$$

연속방정식에서 비압축성 조건의 경우, $\nabla \cdot V = 0$는 유동함수 항을 더하면,

$$\nabla \cdot V = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial z} = \frac{\partial}{\partial x}\left(\frac{\partial \Psi}{\partial z}\right) + \frac{\partial}{\partial z}\left(-\frac{\partial \Psi}{\partial x}\right) = \frac{\partial^2 \Psi}{\partial x \partial z} - \frac{\partial^2 \Psi}{\partial z \partial x} = 0 \qquad [2.5]$$

또한 비회전 조건에서부터,

$$\nabla \times V = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial z} = \frac{\partial}{\partial x}\left(-\frac{\partial \Psi}{\partial x}\right) - \frac{\partial}{\partial z}\left(\frac{\partial \Psi}{\partial z}\right) = \frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial z^2} = 0 \qquad [2.6]$$

2.2. 기본 적용 공식

Green의 identity에 의해서 위의 $\nabla^2 \Phi^* = 0$에 대한 일반적인 해는 식 [2.7]과 같다.

$$\Phi^*(x, y, z) = \left(\frac{-1}{4\pi}\right) \int_{S_B} \left[\sigma\left(\frac{1}{r}\right) - \mu n \cdot \nabla\left(\frac{1}{r}\right)\right] dS + \Phi_\infty \qquad [2.7]$$

여기에서 n은 포텐셜 jump μ의 방향이다. Φ_∞는 자유흐름 포텐셜이며 식 [2.8]과 같이 표기된다.

$$\Phi_\infty = U_\infty x + V_\infty y + W_\infty z \qquad [2.8]$$

경계조건이 없어 많은 해가 나올 수 있다. 다시 적당한 용출과 중첩강도의 조합을 구하려면 후류를 도입하여 해를 식 [2.9]와 같이 표현할 수 있다.

$$\Phi^*(x, y, z) = \left(\frac{1}{4\pi}\right) \int_{body+wake} \mu n \cdot \nabla\left(\frac{1}{r}\right) dS - \frac{1}{4\pi} \int_{body} \sigma\left(\frac{1}{r}\right) dS + \Phi_\infty \qquad [2.9]$$

2.3. Neumann 경계 조건

경계치 문제에서 계의 경계에 대한 부가적인 조건이 도함수 형태로 주어졌을 때의 조건이며, 벽 경계 S_B에서 다음을 만족시켜야 한다. $\frac{\partial \Phi^*}{\partial n}$ 을 구체화 시키면,

$$V \cdot n = (\nabla (\Phi + \Phi_\infty)) \cdot n = 0 \qquad [2.10]$$

즉 경계의 수직한 방향으로는 속도성분이 0이다.

2.4. 무한 경계조건

경계 외부, 먼 지점에서 흐름은 층류를 유지함으로,

$$\lim_{r \to \infty} \nabla \Phi = 0, \text{ 여기에서 } r = (x,y,z) \qquad [2.11]$$

해에 노이만 경계조건을 적용하려면 ∇을 적용하여 식 [2.12]와 같이 변형시킨다.

$$\nabla \Phi^*(x,y,z) = \left(\frac{-1}{4\pi}\right) \int_{body+wake} \mu \nabla \left[\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right] dS - \frac{1}{4\pi} \int_{body} \sigma \ \nabla\left(\frac{1}{r}\right) dS + \nabla \Phi_\infty \qquad [2.12]$$

노이만 조건에 적용하면 식 [2.13]과 같다.

$$\left\{\left(\frac{-1}{4\pi}\right) \int_{body+wake} \mu \nabla \left[\frac{\partial}{\partial n}\left(\frac{1}{r}\right)\right] dS - \frac{1}{4\pi} \int_{body} \sigma \ \nabla\left(\frac{1}{r}\right) dS + \nabla \Phi_\infty\right\} \cdot n = 0 \qquad [2.13]$$

이 식은 많은 수치해의 기초가 되며 경계면의 각 점에서 성립하여야 한다.

또한 경계면 내부에서 특이점은 변하지 않는다.

$$\Phi_i^* = const \qquad [2.14]$$

여기에서 상수는 0으로 대치될 수 있다. 위의 두 경계조건, 즉 노이만 경계조건과 무한 경계조건으로서 Neumann Exterior 문제를 성립하며, 비압축성, 비점성 지배 방정식은 Green의 정리와 부수적인 Kutta 조건을 적용함으로서 해를 구할 수 있다.

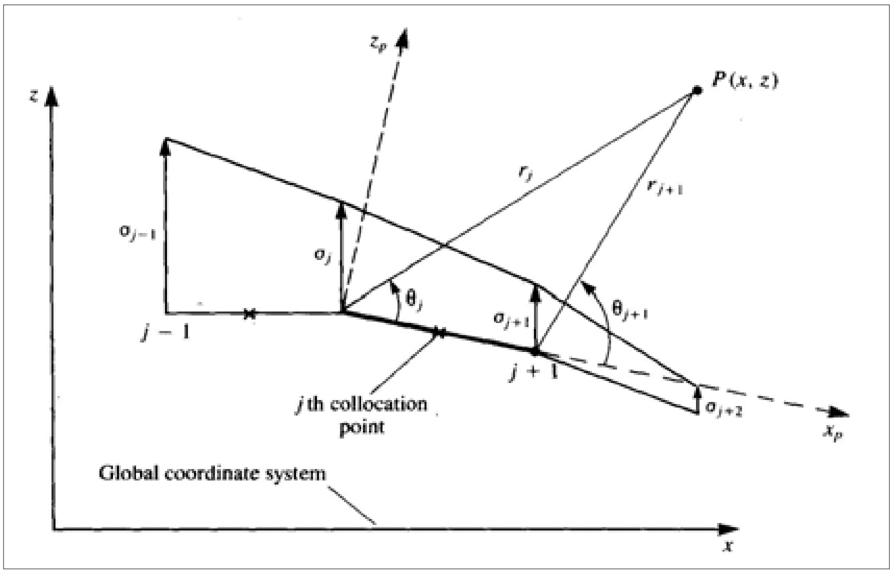

Fig. 2 Layout for LSVP method (Ref.[6])

2.5. Linear Strength Vortex Panel Method(LSVP Method)

Fig.2는 LSVP를 적용하기 위해 필요한 기하학적 표시를 보여주고 있다. 다음은 Linear Strength Vortex Panel Method를 적용하기 위한 단계이다.(Ref.[6])

2.5.1. 특이요소 선택

여러 단계를 통한 속도성분은 다음과 같다.

$$u_p = \frac{\gamma_o}{2\pi}\left[\tan^{-1}\frac{z}{x-x_2} - \tan^{-1}\frac{z}{x-x_1}\right] + \frac{\gamma_1}{4\pi}\left[z\ln\frac{(x-x_1)^2+z_2}{(x-x_2)^2+z_2} + 2x\left(\tan^{-1}\frac{z}{x-x_2} - \tan^{-1}\frac{z}{x-x_1}\right)\right] \quad [2.15]$$

$$w_p = \frac{-\gamma_0}{4\pi}\ln\frac{(x-x_1)^2+z^2}{(x-x_2)^2+z^2} - \frac{\gamma_1}{2\pi}\left[\frac{x}{2}\ln\frac{(x-x_1)^2+z^2}{(x-x_2)^2+z^2} + (x_1-x_2) + z\left(\tan^{-1}\frac{z}{x-x_2} - \tan^{-1}\frac{z}{x-x_1}\right)\right] \quad [2.16]$$

하첨자 1과 2는 패널 모서리 j 와 j+1을 나타낸다. 위 식을 γ, γ_{j+1}에 대한 식으로 변형시키면 다음과 같다.

$$u_p=\frac{z}{2\pi}\left(\frac{\gamma_{j+1}-\gamma_j}{x_{j+1}-x_j}\right)\ln\frac{r_{j+1}}{r_j}+\frac{\gamma_j(x_{j+1}-x_j)+(\gamma_{j+1}-\gamma_j)(x-x_j)}{2\pi(x_{j+1}-x_j)}(\theta_{j+1}-\theta_j) \quad [2.17]$$

$$\begin{aligned}w_p=&-\frac{\gamma_j(x_{j+1}-x_j)+(\gamma_{j+1}-\gamma_j)(x-x_j)}{2\pi(x_{j+1}-x_j)}\ln\frac{r_j}{r_{j+1}}\\&+\frac{z}{2\pi}\left(\frac{\gamma_{j+1}-\gamma_j}{x_{j+1}-x_j}\right)\left[\frac{(x_{j+1})}{z}+(\theta_{j+1}-\theta_j)\right]\end{aligned} \quad [2.18]$$

개념적으로,

$$\begin{pmatrix} u & w \\ u^a & w^a \\ u^b & w^b \end{pmatrix}_{ij} = VOR2DL(\gamma_j,\ \gamma_{j+1},\ x_i,\ z_i,\ x_j,\ z_j,\ x_{j+1},\ z_{j+1}) \quad [2.19]$$

$$(u,w)=(u^a,w^a)+(u^b,w^b) \quad [2.20]$$

2.5.2. 형상이산화

두께비가 두꺼운 에어포일의 경우 LE와 TE에서 패널 밀도를 조밀하게 형성한다. x좌표의 간격은 코사인 간격 공식을 적용한다.

$$x=\frac{c}{2}(1-\cos\beta) \quad [2.21]$$

x 축이 일정 강도를 가지고 있는 N개의 패널로 구성되면, N+1의 모서리 포인트가 계산되며 이를 통해 제어점과 제어점의 값 및 제어점들의 법선 벡터가 계산된다.

2.5.3. 영향 계수

이 단계에서는 0 법선 유동 경계조건이 적용된다. 켐버선에서 법선 속도성분은 자기유도(Self-Induced) 속도와 자유흐름 속도로 구성된다. 그러므로 표면에서 식[2.22]는,

$$q\cdot n=0 \quad [2.22]$$

식[2.23]으로 분해된다.

$$(u,w)\cdot n+(U_\infty, W_\infty)\cdot n=0 \quad [2.23]$$

구체적으로 단위 강도 γ_j와 γ_{j+1}를 가지고 있는 j 번째 요소에 의한 자기유도(Self-Induced) 속도는 처음 제어점에서 식[2.19]를 사용하여 구한다.

$$\begin{pmatrix} u & w \\ u^a & w^a \\ u^b & w^b \end{pmatrix}_{1j} = VOR2DL(\gamma_j=1,\ \gamma_{j+1}=1,\ x_i,\ z_i,\ x_j,\ z_j,\ x_{j+1},\ z_{j+1}) \quad [2.24]$$

각 제어점에 속도는 j번째 패널의 두 모서리에 의해 영향을 받는다. 그러므로 j+1 번째 패널과 다음 패널의 영향을 더하면 첫 번째 제어점에서 국부 자기유도(Self-Induced) 속도를 구할 수 있다.

$$\begin{aligned}(w, w)_1 = (u^a, w^a)_{11}\gamma_1 &+ [(u^b, w^b)_{11} + (u^a, w^a)_{12}]\gamma_2 \\ &+ [(u^b, w^b)_{12} + (u^a, w^a)_{13}]\gamma_3 + \ldots \\ &+ [(u^b, w^b)_{1,N-1} + (u^a, w^a)_{1N}]\gamma_N + (u^b, w^b)_{1N}\gamma_{N+1}\end{aligned} \quad [2.25]$$

이 방정식은 줄여서,

$$(u, w)_1 = (u, w)_{11}\gamma_1 + (u, w)_{12}\gamma_2 + \ldots + (u, w)_{1,N+1}\gamma_{N+1} \quad [2.26]$$

첫 번째와 마지막 항은,

$$\begin{aligned}(u, w)_{11}\gamma_{11} &= (u^a, w^a)_{11}\gamma_1 \\ (u, w)_1\gamma_{N+1} &= (u^b, w^b)_{1N}\gamma_{N+1}\end{aligned} \quad [2.27]$$

모든 다른 항들은 식 [2.28]과 같이 표현된다.

$$(u, w)_{1,j} = [(u^b, w^b)_{1,j-1} + (u^a, w^a)_{1,j}]\gamma_j \quad [2.28]$$

$\gamma_j = 1$일시 영향 계수가 구해진다.

$$a_{ij} = (u, w)_{i,j} \cdot n_i \quad [2.29]$$

각각의 제어점에는 N+1의 계수와 미지수 γ_j가 있다.

2.5.4. 경계조건 성립

법선 자유흐름 속도 성분 RHS_i는 이산 와도 경우에 구해질 수 있다.

$$RHS_i = -(U_{\infty,} W_{\infty}) \cdot (\cos\alpha_{i,} - \sin\alpha_i) \tag{2.30}$$

2.5.5. 방정식의 해

각각의 제어점(i=1 -> N)에 대한 경계조건은 미지수 γ_j(j=1 -> N+1)을 가지고 있는 N개의 선형대수 방정식으로 나타난다. 추가적인 방정식은 TE에 대해 Kutta 조건을 적용함으로서 구해질 수 있다.

$$\gamma_1 + \gamma_{N+1} = 0 \tag{2.31}$$

그러면 식 [2.32]와 같은 형태를 얻는다.

$$\begin{pmatrix} a_{11} & a_{12} & & \cdots & & a_{1,N+1} \\ a_{21} & a_{22} & & \cdots & & a_{2,N+1} \\ a_{31} & a_{32} & & \ddots & & a_{3,N+1} \\ \vdots & \vdots & & & & \vdots \\ a_{N1} & a_{N2} & & \cdots & & a_{N,N+1} \\ 1 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_N \\ \gamma_{N+1} \end{pmatrix} = \begin{pmatrix} RHS_1 \\ RHS_2 \\ RHS_3 \\ \vdots \\ RHS_N \\ 0 \end{pmatrix} \tag{2.32}$$

2.5.6. 압력과 하중 계산

와도 γ_j의 강도가 알려진 후 각각의 제어점에서 교란속도는 와도 분포의 결과를 이용하여 계산될 수 있다.

$$Q_{tj} = (Q_{t\infty})_j + \frac{\gamma_j + \gamma_{j+1}}{4}, \quad C_p = 1 - \frac{Q_t^2}{Q_\infty^2} \tag{2.33}$$

패널의 양력은 압력 분포 내지 Kutta-Joukowsky 이론을 적용하여 계산될 수 있다.

$$\Delta L_j = \rho Q_\infty \frac{\gamma_j + \gamma_{j+1}}{2} \Delta c_j \tag{2.34}$$

Δc_j는 패널 길이이다. 총 양력과 모멘트는 요소를 더함으로써 구할 수 있다.

2.6. 힘분포 및 공력계수

다음 그림은 비행기 날개에 분포되는 수직 및 수평 응력 분포에 따른 힘의 관계를 보여준다(Ref.[4]).

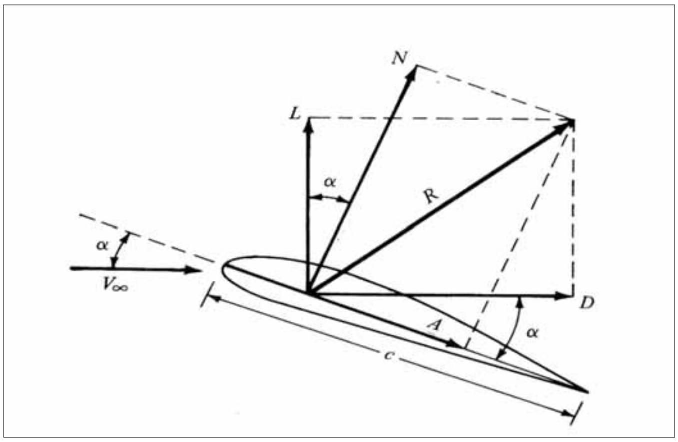

Fig. 3 Force distribution (Ref.[4])

상단면에는, $dN_u' = -p_u ds_u \cos\theta - \tau_u ds_u \sin\theta$ [2.35]

$dA_u' = -p_u ds_u \sin\theta + \tau_u ds_u \cos\theta$ [2.36]

하단면에는, $dN_l' = p_l ds_l \cos\theta - \tau_l ds_l \sin\theta$ [2.37]

$dA_l' = p_l ds_l \sin\theta + \tau_l ds_l \cos\theta$ [2.38]

각 방향별로 단위 스팬당 힘으로 분해하면 다음과 같다.

$$N' = -\int_{LE}^{TE} (p_u \cos\theta + \tau_u \sin\theta) ds_u + \int_{LE}^{TE} (p_l \cos\theta - \tau_l \sin\theta) ds_l \quad [2.39]$$

$$A' = \int_{LE}^{TE} (-p_u \sin\theta + \tau_u \cos\theta) ds_u + \int_{LE}^{TE} (p_l \sin\theta + \tau_l \cos\theta) ds_l \quad [2.40]$$

이 힘으로 인한 LE에 대한 미소 모멘트는 다음과 같다.

$$dM'_u = (p_u \cos\theta + \tau_u \sin\theta) x \; ds_u + (-p_u \sin\theta + \tau_u \cos\theta) y \; ds_u \quad [2.41]$$

$$dM'_l = (-p_l \cos\theta + \tau_l \sin\theta) x \; ds_l + (p_l \sin\theta + \tau_l \cos\theta) y \; ds_l \quad [2.42]$$

$$M_{LE}' = \int_{LE}^{TE} [(p_u \cos\theta + \tau_u \sin\theta) x + (-p_u \sin\theta + \tau_u \cos\theta) y] \; ds_u \quad [2.43]$$

$$+ \int_{LE}^{TE} [(-p_l \cos\theta + \tau_l \sin\theta) x + (p_l \sin\theta + \tau_l \cos\theta) y] \; ds_l \quad [2.44]$$

실질적으로는 측정이 불가능한 부분이 있으므로 차원해석에 의한 힘과 모멘트가 정의되며 2-D 경우 다음과 같다.

$$c_l \equiv \frac{L'}{q_\infty c}, \quad c_d \equiv \frac{D'}{q_\infty c}, \quad c_m \equiv \frac{M'}{q_\infty c^2}, \quad C_p \equiv \frac{p-p_\infty}{q_\infty}, \quad c_f \equiv \frac{\tau}{q_\infty} \quad [2.45]$$

위 계수를 언급된 식에 적용하면 식 [2.46], 식 [2.47], 식 [2.48]과 같다.

$$c_n = \frac{1}{c}\left[\int_0^c (C_{p,l} - C_{p,u})dx + \int_0^c \left(c_{f,u}\frac{dy_u}{dx} + c_{f,l}\frac{dy_l}{dx}\right)dx\right] \quad [2.46]$$

$$c_a = \frac{1}{c}\left[\int_0^c \left(C_{p,u}\frac{dy_u}{dx} - C_{p,l}\frac{dy_l}{dx}\right)dx + \int_0^c (c_{f,u} + c_{f,l})dx\right] \quad [2.47]$$

$$c_{m_{LE}} = \frac{1}{c^2}\left[\int_0^c (C_{p,u} - C_{p,l})x\,dx + \int_0^c \left(c_{f,u}\frac{dy_u}{dx} + c_{f,l}\frac{dy_l}{dx}\right)x\,dx + \int_0^c \left(C_{p,u}\frac{dy_u}{dx} + c_{f,u}\right)y_u\,dx + \int_0^c \left(-C_{p,l}\frac{dy_l}{dx} + c_{f,l}\right)y_l\,dx\right] \quad [2.48]$$

2.7. 압력중심

Fig. 4에서 모멘트에 관해 정리하면 $M'_{LE} = -(x_{cp})N'$ 이며 x_{cp}는 압력의 중심으로 정의된다. 만일 모멘트가 압력중심에 대해 정의된다면 분산 압력의 합은 0이 된다. 그림에서 평형축 정리를 사용한다면 식 [2.49]와 같이 표현된다.

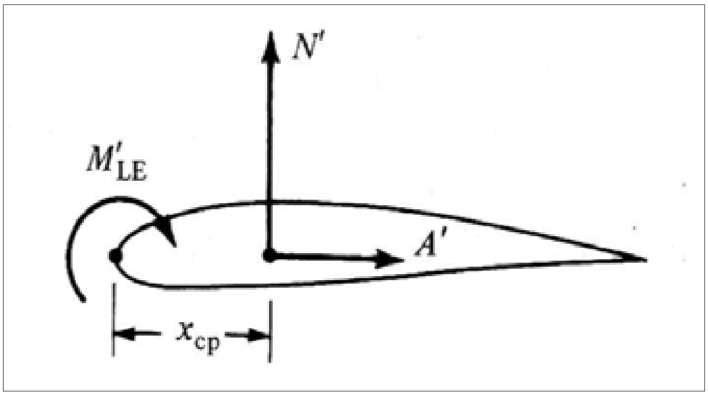

Fig. 4 Moment and Center of Pressure (Ref.[4])

$$M'_{LE} = -\frac{c}{4}L' + M'_{c/4} = -x_{cp}L' \quad [2.49]$$

2.8. 공력중심

공력중심의 정의는 공기역학적으로 발생된 모멘트가 받음각에 대해 무관하게 되는 점을 의미한다. 공력중심은 코드의 1/4지점에 근접하는 경향을 보인다. 공력중심을 Fig. 5에서와 같이 cx_{ac}로 표시하고 이 점에 대해 모멘트를 구하면,

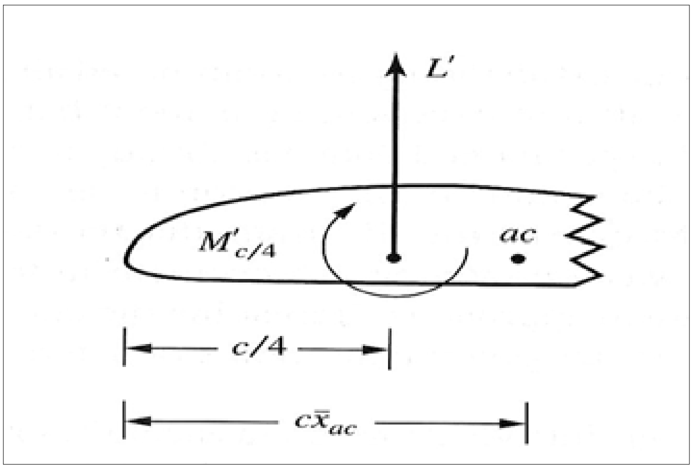

Fig. 5 A quarter point Cm and Xac(Ref.[4])

$$M'_{ac} = L'(cx_{ac} - c/4) + M'_{c/4} \quad [2.50]$$

양변을 $q_\infty Sc$로 나누어 계수형태로 변환하면,

$$\frac{M'_{ac}}{q_\infty Sc} = \frac{L'}{q_\infty S}(x_{ac} - 0.25) + \frac{M'_{c/4}}{q_\infty Sc} \quad [2.51]$$

$$c_{m,ac} = c_l(x_{ac} - 0.25) + c_{m,c/4} \quad [2.52]$$

Ref.[4]에 의해 결론적으로 공력 중심에 대해 식 [2.53]과 같이 유도된다.

$$x_{ac} = -\frac{m_0}{a_0} + 0.25 \quad [2.53]$$

2.9. LE에 대한 모멘트

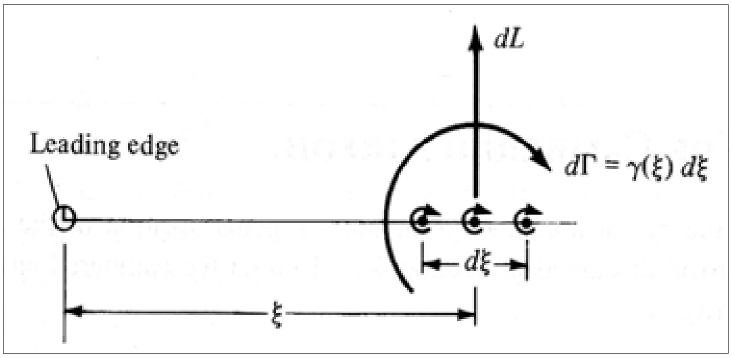

Fig. 6 Moment about leading edge (Ref.[4])

여러 변환을 통해 모멘트 계산을 위해 주어지는 식은 식 [2.54]와 같다. 여기에서 A_0와 A_1은 Ref. [4]의 p.307에서 구해진다.

$$c_{mle} = -\frac{\pi}{2}\left(A_0 + A_1 - \frac{A_2}{2}\right) \qquad [2.54]$$

$c_l = \pi(2A_0 + A_1)$를 대입하면,

$$c_{mle} = -\left[\frac{c_l}{4} + \frac{\pi}{4}(A_1 - A_2)\right] \qquad [2.55]$$

1/4 지점에서 모멘트는,

$$c_{m,\,c/4} = c_{mle} + \frac{c_l}{4} \qquad [2.56]$$

정리하면,

$$c_{m,\,c/4} = \frac{\pi}{4}(A_2 - A_1) \qquad [2.57]$$

식 [2.57]이 알려주는 의미는 1/4 지점이 압력중심이 아니라는 것, 또한 A_1, A_2는 캠버선의 기하학적 모양에 관계있다는 것, 그리고 $c_{m,\,c/4}$는 받음각에 무관하다는 거이다. 이전에 정의된 압력중심을 적용하면,

$$x_{cp} = -\frac{M'_{LE}}{L'} = -\frac{c_{mle}c}{c_l} \qquad [2.58]$$

$$x_{cp} = \frac{c}{4}\left[1 + \frac{\pi}{c_l}(A_1 - A_2)\right] \qquad [2.59]$$

식 [2.59]는 캠버 에어포일 경우 양력계수에 따라 압력중심이 변하는 것을 보여준다. 그러므로 받음각이 변하면 압력중심(Center of Pressure)도 변한다.

2.10. 스플라인

Ref. [8]에 의하면 스플라인은 다음과 같다. 다항식의 차수가 큰 경우에는 주어진 구간을 여러 개의 소구간으로 나누고, 소구간 별로 보다 낮은 차수의 다항식을 사용하여 데디터 점들을 지나는 매끈한 곡선을 얻는 방법을 Spline 보간법이라고 한다. 이중 3차 다항식을 사용한 spline을 3차 spline 보간법(cubic spline interpolation)이라고 한다. 구간[$x_{0,} x_n$]에서 함수 f(x)가 주어질 때 이 구간을 여러 개의 소구간에서 3차 다항식으로 f(x)를 근사화 한다. 소구간 [x_i, x_{i+1}]에서 구하고자 하는 3차 다항식을 식 [2.60]과 같이 가정한다.

$$f_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + b_i(x-x_i) + d_i, \qquad i = 0.1.2. \ldots, n-1 \qquad [2.60]$$

소구간에서 4개의 미지상수를 구해햐 하므로 n개의 소구간에 대하여 4n의 조건이 필요하다. 알 수 있는 조건은, 첫 번째로 다항식은 수구간 양 끝점을 지난다.

$$f_i(x_i) = y_{i,} \qquad i = 0,1,2\ldots,n-1 \qquad [2.61]$$

$$f_i(x_{i+1}) = y_{i,} \qquad i = 0,1,2\ldots,n-1 \qquad [2.62]$$

두 번째로 각 소구간의 다항식들은 그들이 만나는 양 끝점에서 1계 도함수와 2계 도함수는 같아야 한다.

$$f_{i-1}{}'(x_i) = f_i'(x_i), \qquad i = 0,1,2\ldots,n-1 \qquad [2.63]$$

$$f_{i-1}{}''(x_i) = f_i''(x_i), \qquad i = 0,1,2\ldots,n-1 \qquad [2.64]$$

첫 번째 조건을 적용하면,

$$y_i = d_i \qquad [2.65]$$

$$y_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \qquad [2.66]$$

점 (x_i, y_i)와 (x_{i+1}, y_{i+1})에서 2계 도함수를 각각 σ_i와 σ_{i+1}로 놓으면 기본 다항식에서 다음의 식 [2.67]과 같다.

$$\sigma_i = 2b_i, \qquad \sigma_{i+1} = 6a_i h_i + 2b_i \qquad [2.67]$$

위 식 [2.65], 식 [2.66]에서,

$$b_i = \frac{\sigma_i}{2}, \quad a_i = \frac{\sigma_{i+1} - \sigma_i}{6h_i}, \quad c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i\sigma_i + h_i\sigma_{i+1}}{6} \qquad [2.68]$$

모든 계수를 대입하면 소구간[x_i, x_{i+1}]에서 3차 다항식은 [2.69]와 같다.

$$f_i(x) = \frac{\sigma_i}{6}\left\{\frac{(x_{i+1}-x)^3}{h_i} - h_i(x_{i+1}-x)\right\} + \frac{\sigma_{i+1}}{6}\left\{\frac{(x-x_i)^3}{h_i} - h_i(x-x_i)\right\}$$
$$+ y_i\left\{\frac{x_i+1-x}{h_i}\right\} + y_{i+1}\left\{\frac{x-x_i}{h_i}\right\}, \qquad i=1,2,\ldots,n-1 \qquad [2.69]$$

식 [2.69]에서 σ_i와 σ_{i+1}의 값을 알면, 소구간 [x_i, x_{i+1}]에서 함수 $f_i(x)$를 결정할 수 있다. 그러므로 (n+1)개의 2계 도함수,

$$\sigma_0, \sigma_1, \ldots, \sigma_n \qquad [2.70]$$

$$f_i'(x) = \frac{\sigma_i}{6}\left\{\frac{-3(x_{i+1}-x)^2}{h_i} + h_i\right\} + \frac{\sigma_{i+1}}{6}\left\{\frac{-3(x-x_i)^2}{h_i} - h_i\right\} + \Delta y_i \qquad [2.71]$$

$$f_i'(x_i) = \frac{\sigma_i}{6}(-2h_i) + \frac{\sigma_{i+1}}{6}(-h_i) + \Delta y_i \qquad [2.72]$$

$$f_{i-1}'(x_i) = \frac{\sigma_{i-1}}{6}(h_{i-1}) + \frac{\sigma_i}{6}(2h_{i-1}) + \Delta y_{i-1}, \text{ 여기에서 } \Delta y_i = \left(\frac{y_{i+1}-y_i}{h_i}\right) \qquad [2.73]$$

$x = x_i$에서 좌우측 1계 도함수는 같아야 하므로 위의 1계 도함수를 같게 하고 정리하면 최종식 [2.74]을 얻을 수 있다.

$$h_{i-1}\sigma_{i-1} + 2(h_{i-1}+h_i)\sigma_i + h_i\sigma_{i+1} = 6(\Delta y_i - \Delta y_{i-1}), \qquad i=1,2,3,\ldots,n-1 \qquad [2.74]$$

양 끝점에서 처리는 2계 도함수를 0으로 놓아 함수를 직선으로 가정한다. 이는 일반 사인 코사인 곡선에서 x 절편에서 2계 도함수가 0으로 놓인 것과 비슷한 의미를 갖는다고 가정한다.

2.11. NURB 곡선

NURB는 'Non-Uniform Rational B-spline'이며 B-spline은 자체가 non-uniform이다. Rational은 아핀 공간(Affine Space)로 사상시켰으므로 사용한다. B-spline 곡선은 여러 다항식 세그먼트로 이루어진 곡선으로 차수를 제한하지 않으나 각 세그먼트간 차수는 같아야 한다. B-spline 곡선은 각 다항식 세그먼트의 n차의 차수와 세그먼트의 수 L에 의해 정의된다. 또한 곡선의 domain도 필요하다. 이러한 partition은 knot u_i의 비감소 시퀀스이다. 매개변수 u가 도메인을 따라가면서 B-spline 곡선상 상응점 d(u)은 제어점의 집합인 d_i로 정의된 다항식 곡선을 따라간다. 이 곡선의 매끈함은 knots 시퀀스 구성의 정확성에 의해 결정된다.

NURB 기본형은 식 [2.75]와 같으며, 구간 $\lambda=[u_I, u_I+1]$에 대해 $u\in\lambda$라고 하면,

$$S(u)=\sum_{i=0}^{p} d_i N_i^m(u); \quad d_i\in P^3 \tag{2.75}$$

N_i^m은 부분적 다항식이며 다음 식 [2.76]의 recursion을 만족시킨다.

$$N_l^m(u)=\frac{u-u_{l-1}}{u_{l+n-1}-u_{l-1}} N_l^{m-1}(u)+\frac{u_{l+n}-u}{u_{l+n}-u_l} N_{l+1}^{m-1}(u) \tag{2.76}$$

Recursion 조건은,

$$N_i^0(u)=\begin{cases}1 & \text{if } u_{i-1}\le u< u_i \\ 0 & else\end{cases} \tag{2.77}$$

affine 공간에서, explicit function 형태는 rational B-spline에 대응된다.

$$S(u)=\frac{\sum_{i=0}^{p} w_i d_i N_i^m(u)}{\sum_{i=0}^{p} w_i N_i^m(u)}; \quad d_i\in P^3. \tag{2.78}$$

간단하게,

$$S(u)=\sum_{i=0}^{p} d_i R_i^m(u), \quad \text{여기에서} \quad R_i^m(u)=\frac{w_i N_i^m(u)}{\sum_{j=0}^{p} w_j N_j^m(u)} \tag{2.79}$$

3. 결과 및 고찰(Result)

3.1. 흐름도

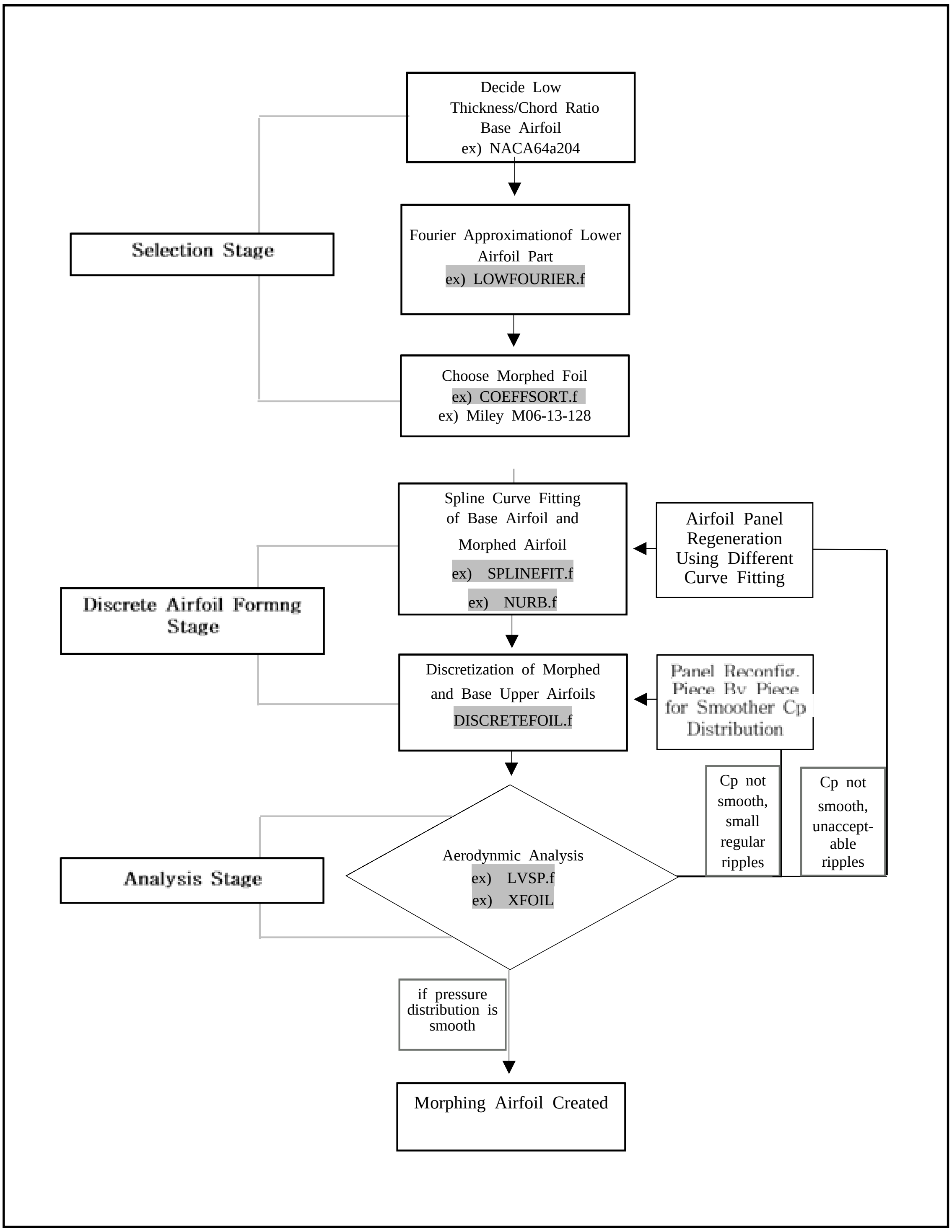

Fig. 7 Schematic chart showing work flow, in brevity

3.2. 접근방법

3.2.1. 푸리에 적합에 의한 에어포일 선택

초기 작업(Preliminary Approach)의 일환으로, 기준인 고 T/C비 에어포일의 하반면과 같은 저 T/C비 에어포일의 하반면을 구하려면 최소 기하학적 일치성 내지, 적어도 제한된 오차 이내로 들어오는 에어포일 곡선을 구하여야 한다. 임의 곡선은 사인과 코사인의 합으로 표현될 수 있다는 가정하에 Fourier 근사법으로 주어진 에어포일 하반면의 곡선에 근사시켜 이를 기준으로 추세를 확인한다.

주기를 가지고 있는 임의 곡선은 식 [3.1]처럼 표현될 수 있다.

$$f(t) = a_0 + c\cos(\omega t + \phi) \qquad [3.1]$$

c는 진폭이며 a_0는 그래프의 시작점이다. 여기에서 에어포일의 ω를 π(rad/s)로 가정한다. 이유는 에어포일 하반면을 그래프를 한 주기의 반으로 볼 수 있기 때문이다. 표현하면,

$$f(t) = a_0 + a_1\cos(\omega t) + b_1\sin(\omega t) \qquad [3.2]$$

$$a_1 = c\cos\phi, \qquad b_1 = -c\sin\phi \qquad [3.3]$$

$$c = \sqrt{(a_0)^2 + (a_1)^2} \qquad [3.4]$$

정현파에 대한 근사적인 표현인 최소자승법은 식 [3.5]와 같다. 주어진 다점 $(t_{1,} Y_1)$, $(t_{2,} Y_2)$,..., $(t_{N,} Y_N)$에 대해,

$$S = \sum_{i=1}^{N}(Y_i - y_i)^2 = \sum_{i=1}^{N}[Y_i - (a_0 + a_1\cos(\omega t_i) + b_1\sin(\omega t_i))]^2 \qquad [3.5]$$

위 S를 각각의 요소에 대해 미분하고 정리하면 식 [3.6] 같은 행렬식으로 전개된다.

$$\begin{bmatrix} N & \sum_{i=1}^{N}\cos(\omega t_i) & \sum_{i=1}^{N}\sin(\omega t_i) \\ \sum_{i=1}^{N}\cos(\omega t_i) & \sum_{i=1}^{N}\cos^2(\omega t_i) & \sum_{i=1}^{N}\cos(\omega t_i)\sin(\omega t_i) \\ \sum_{i=1}^{N}\sin(\omega t_i) & \sum_{i=1}^{N}\sin(\omega t_i)\cos(\omega t_i) & \sum_{i=1}^{N}\sin^2(\omega t_i) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} Y_i \\ \sum_{i=1}^{N} Y_i\cos(\omega t_i) \\ \sum_{i=1}^{N} Y_i\sin(\omega t_i) \end{bmatrix} \qquad [3.6]$$

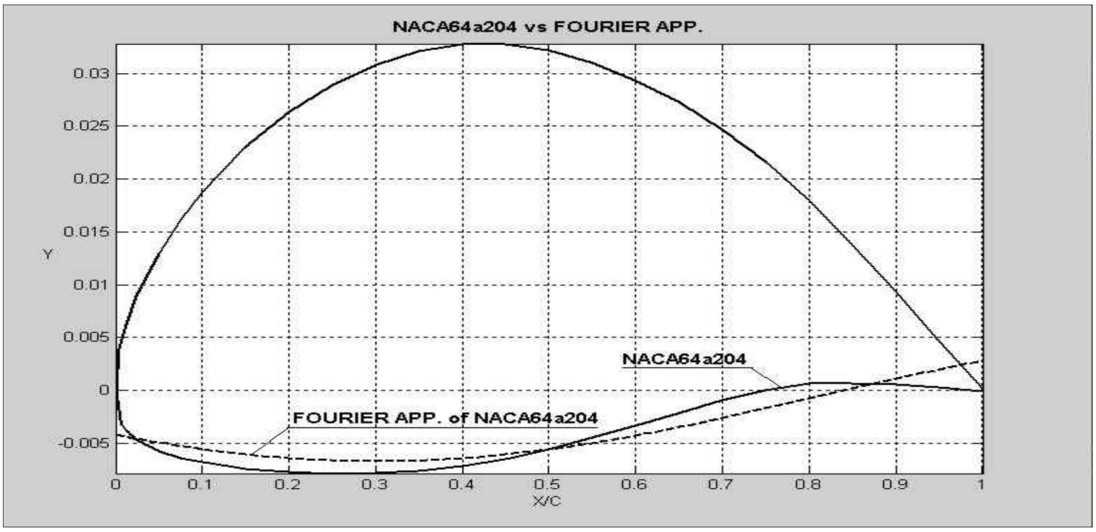

Fig. 8 NACA64a204 and its Fourier approximation, not in 1:1 ratio

에어포일 하반면에 대한 근사적인 적합곡선은 위 행렬식을 통해 계수를 구하는 것으로 귀결된다. 이에 대한 프로그램은 부록에 lowfourier.f로 첨부하였다. 문제는 그 계수에 대한 1548개의 에어포일 파일을 일일이 비교할 수 없을 뿐만 아니라 상호간에 비교되어야 하는 계수가 3개이므로 이를 줄일 필요가 있다. 그러므로 c계수를 위에 언급된바와 같이 구해 c의 크기(magnitude)를 기준으로 파일 자료를 정렬시키며 푸리에 근사 계수 정렬 결과는 정렬 프로그램과 함께 마찬가지로 첨부하였다. 계산 프로그램 오차와 파일 형태에서 결과에 대한 오차가 발생할 수 있으나 이는 자료와 프로그램에서 체계적으로 발생하였으므로 결과 값들은 전반적으로 같은 오프셋을 가질 것이며 이로써 일정한 bias error를 유지하므로 계수 결과 값은 유효하다. NACA64a204의 경우 하반면이 다음과 같은 근사치를 보여주므로 유효성을 확인할 수 있다.

에어포일 선정에 있어서 두가지 조건을 만족시켜야한다. 하나는 계수간 근사성이고 두 번째로 고 thickness/chord ratio이어야 한다. 우선 c계수를 기준으로 정렬된 자료의 근방에서 찾는다. 다음 기준 에어포일의 a_1과 비교될 에어포일의 a_1의 값을 비교해서 같거나 거의 비슷해야한다. 그 다음 기준 에어포일의 a_1, b_1의 부호가 같음을 확인하고 다음 b_1의 크기를 비교한다. 마지막으로 a_0의 크기와 부호를 비교하여 가장 근접하는 에어포일을 선정해야 한다. 추가적으로 변형후 에어포일이 기준 에어포일보다 두께값이 chord에 따라서 전반적으로 더 높은 것을 선택한다.

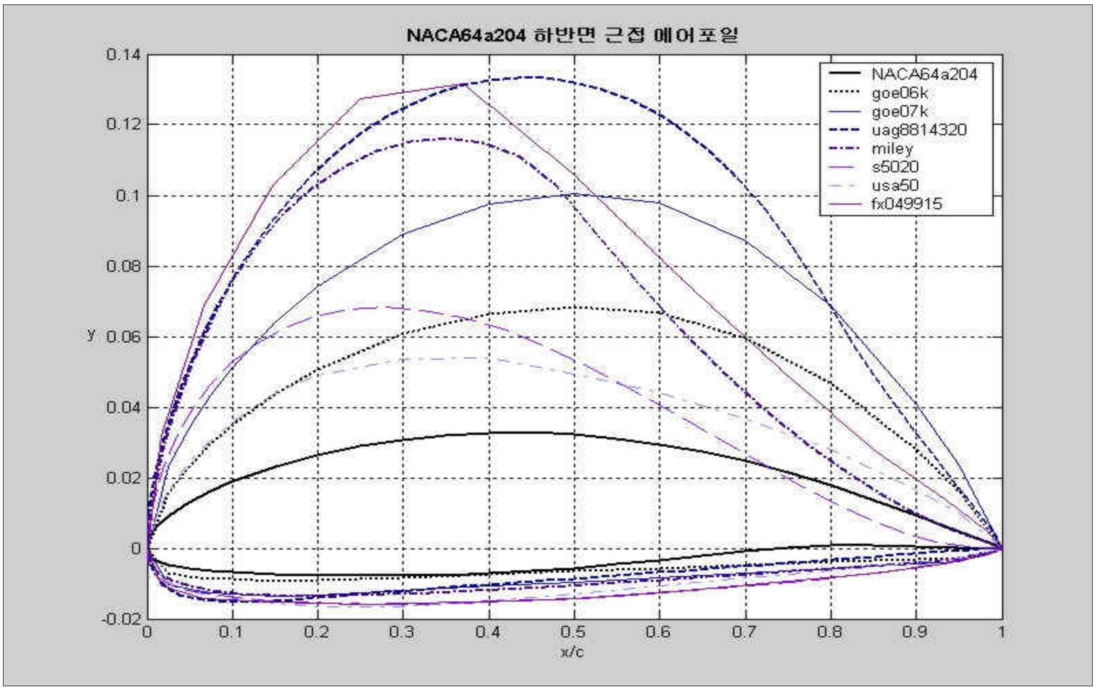

Fig. 9 Lower airfoils close to NACA64a204, not in 1:1 ratio

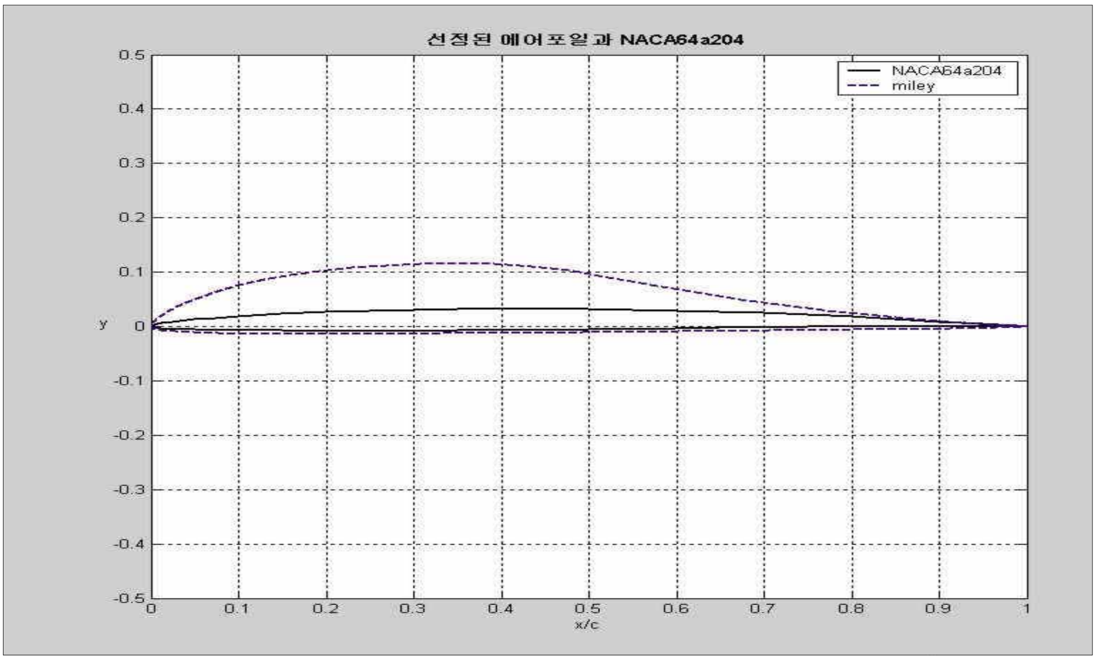

Fig.10 Chosen airfoil(Miley) with NACA64a204 in 1:1 ratio

그 결과 Fig. 9과 같이 여러 에어포일이 나왔다. 이중 가장 적합한 에어포일로서 MILEY M06-13-128을 택하였으며 실제 비율로는 Fig.10 같다. Ref.[5]에서 저속 에어포일로 확인 되었다.

3.2.2 에어포일의 스플라인과 NURB 표현 및 압력 분포

더 정확한 곡선을 표현하기 위해 스플라인 곡선(splinefit.f)을 적용하였으며 Morphing 단계는 0.01의 x축의 등간격으로 Fig.11처럼 나왔다. 그러나 스플라인을 적용한 결과 패널의 기울기가 급격한 구간인 LE 구간에서 압력 물결 현상을 보이고 있으며 이를 보상하기 위해 LE와 TE에서 각 구간에 대해 수작업 다항식 적합을 각각 적용하였다. 공력 선도 전반적으로 압력 분포로는 부적합한 pressure ripple을 보이고 있다. 현상을 보상하기 위해 NURB(Non-Uniform Rational B-splines)곡선(NURB.f)을 적용하여 Fig.13과 같은 패널 분포를 얻었다. NURB를 통한 효과로 LE와 TE 부근에서 생성되는 패널들이 조밀하게 형성하였으며 비교적 기울기 변화가 적은 중간 부분에서는 넓은 패널을 생성하였다. 이러한 패널링이 가능했던 이유는 패널당 많은 knot 삽입(약 1000개)과 NURB로 표현된 하반면과 상반면의 x값 일치여부 비교를 통한 y값 생성의 결과로 생성했기 때문이다.

Fig.12는 NURB 에어포일 자료의 압력곡선을 보여주고 있다. 처음 방법인 일반 스플라인 적합의 경우 LE와 TE에서 아주 급격한 압력 물결을 형성하였다. 보정하기 위해 스플라인 및 구간별 다항식 수작업 경우에는 LE와 TE에서 급격한 압력 변동은 사라졌으나, 미소 압력 물결이 전반적으로 분포하고 있다. 이를 위해 LE와 TE에 대해서 장시간의 다항식 적합을 위한 수작업이 필요했다. NURB를 적용한 경우 LE, TE 추가적인 보정 없이도 하반면과 LE 및 TE에 대해서 Fig.14와 같이 매끈한 압력분포를 얻을 수 있었으나 에어포일 중앙 부분의 제어점을 기점으로 규칙적인 압력 물결을 형성하였다. 패널을 하나씩 양력을 높이는 방향으로 조절하면 임의 받음각에 대한 매끈한 압력분포와 양력 상승을 얻을 수 있을 것이다.

Fig.15는 XFOIL을 사용하여 Miley(upper)-NACA64a204(lower) (이하 Miley-mix) 에어포일에서 받음각의 변화에 대한 Cp 분포도를 보여주고 있다. 특히 XFOIL을 통해 각각의 Morphing 단계에 대해 임의 받음각에 대해 살펴본 결과 부록에 첨부된 것처럼 Stage 2에서 압력 discrepancy가 매우 크게 나왔다.

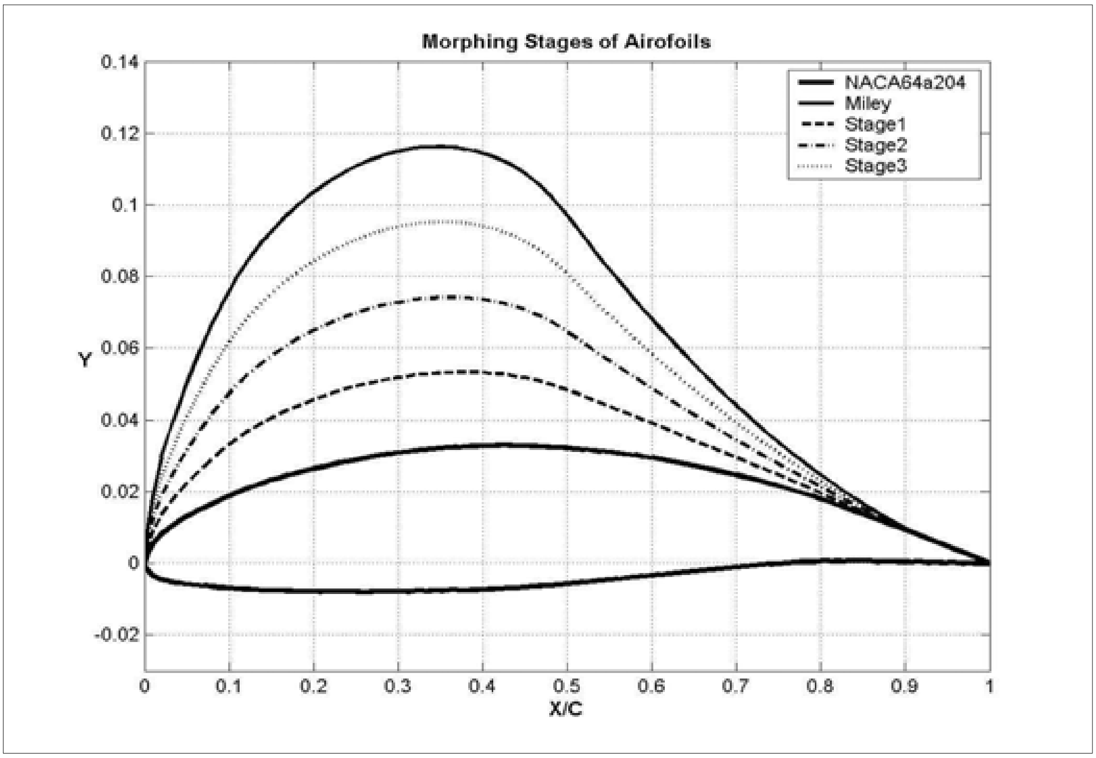

Fig.11 Airfoils created by spline, and piecewise polynomial at LE & TE

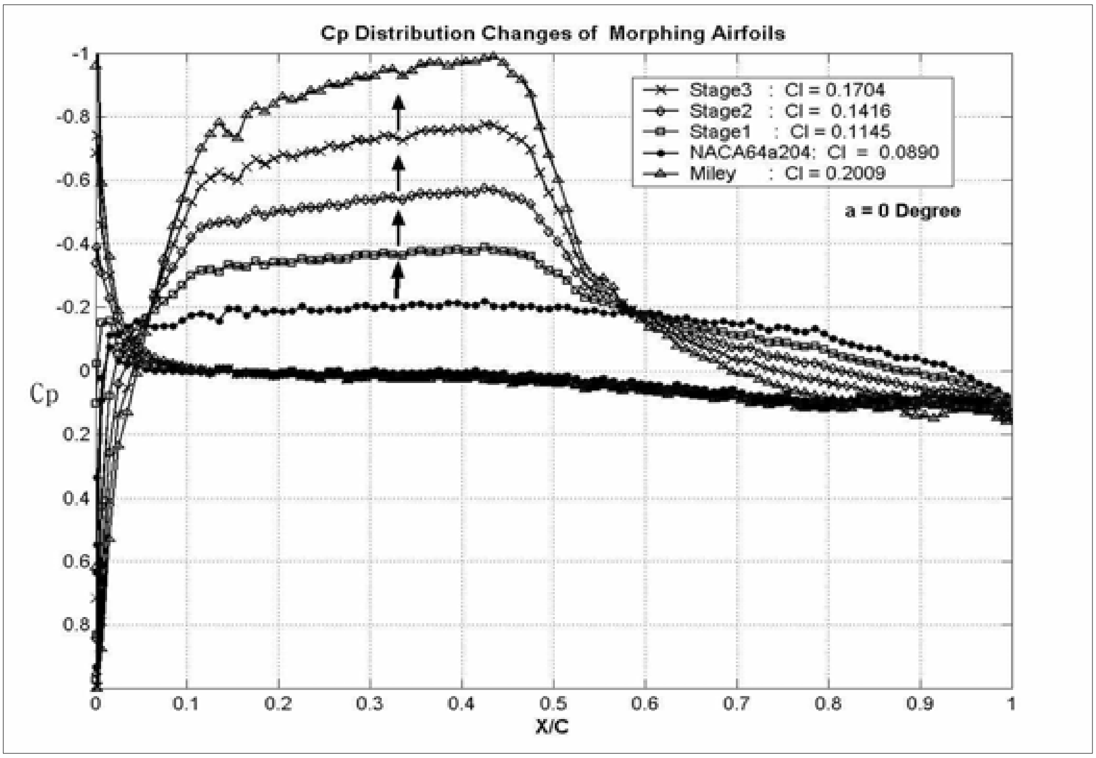

Fig.12 Cp distribution of Fig. 11 for all morphing stages

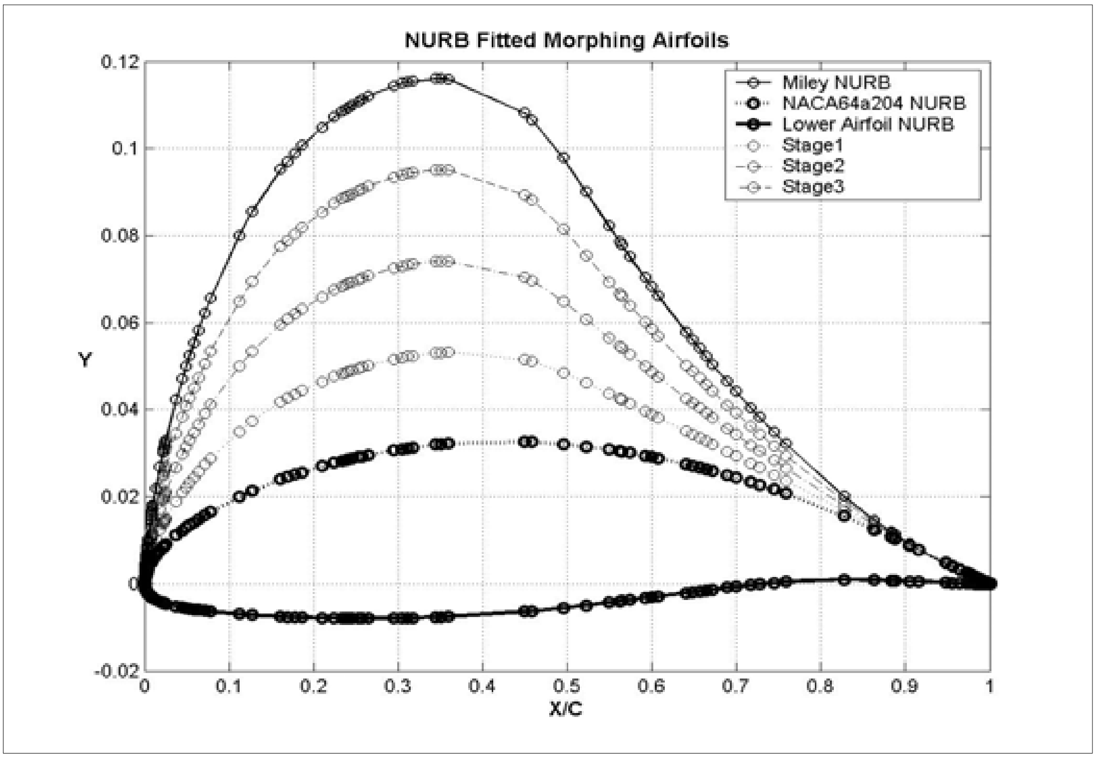

Fig.13 NURB created airfoils for morphing stages

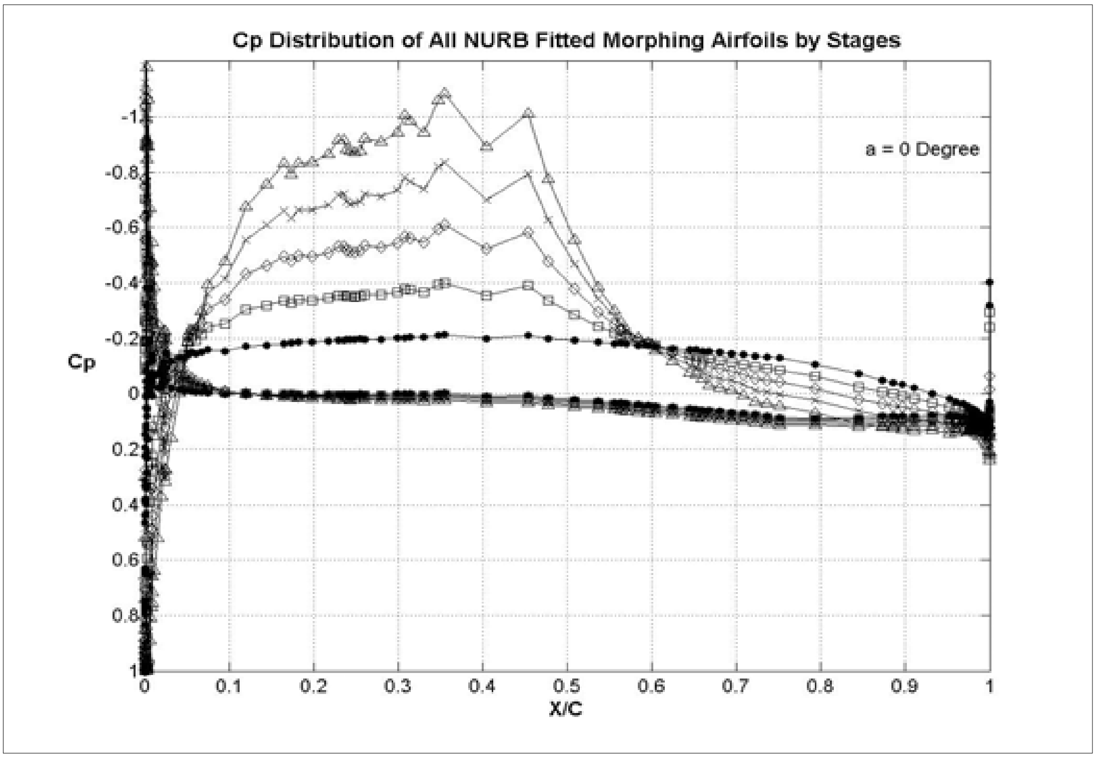

Fig.14 Cp distribution of Fig.13

3.2.3. 패널 재구성

Fig.15에서 보면 두 방법으로 얻은 공력 선도에서 큰 차이는 없었다. 차이에 대한 논의는 제외한다. 다만 x/c 축 범위 0.15~0.45에서 압력 물결이 공통적으로 확인이 되었다. 문제가 되는 부분을 직접 수정하여 보정된 그림은 Fig.16과 같다. 압력 ripple 추세의 확대를 위해 0도에서 20도까지 변화를 주었다. Fig.16 에어포일 중간 부분에서 압력 진동 현상이 현저히 감소되었음을 알 수 있다. 부분적인 재구성에 의해 양력을 높일 수 도 있다.

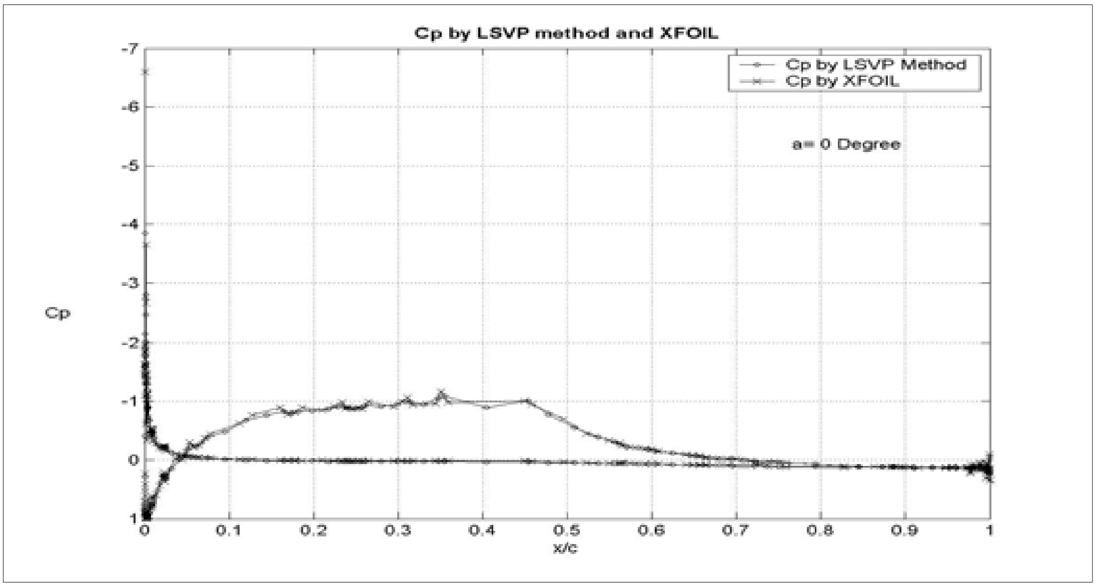

Fig.15 Cp distribution. LSVP and XFOIL on Miley-mix

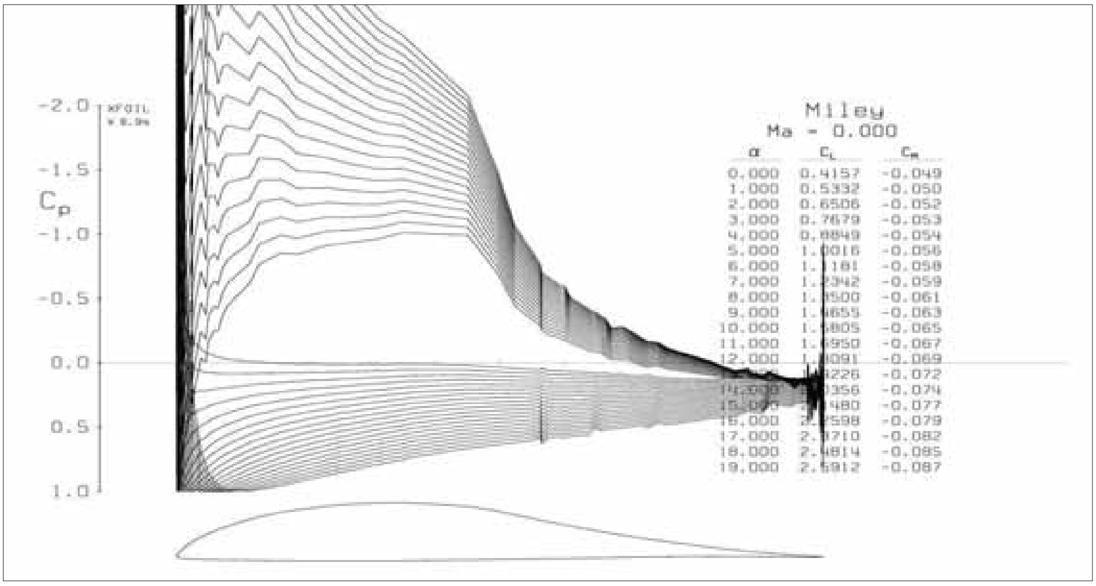

Fig.16 Reconfiguration of Miley-mix airfoil at the mid section

3.2.4. 압력중심 및 공력중심(x_{cp} and x_{ac})에 대한 고찰

정의에 의하면 ‘공력중심은 공기역학적으로 발생된 모멘트가 받음각에 대해 무관하게 되는 점이다.’ 다시 정리하면 임의 받음각에 대해 모멘트 계수가 같은 지점이 공력중심이다. 모멘트 계수는 Ref.[10]에 의해 식 [3.7]과 같다.

$$c_m = -\int_0^c (C_{p,u} - C_{p,l}) x dx \qquad [3.7]$$

이 식에 대한 하나의 에어포일의 결과는 Fig. 17에 나왔다. Ref.[10]에 의한 모멘트 계수 계산식 [3.7]에는 한계를 가지고 있다. 압력 계수는 압력과 동일하며 압력은 따라서 패널에 수직한 방향으로 작용함으로 x방향과 y방향으로 성분이 분해되어 압력 계수의 수평방향 성분도 모멘트 계수 계산에 포함되어야 하나 식에서는 수직방향만 다루고 있다. 위 식에 의한 계산으로는 정확한 압력중심과 공력중심을 얻을 수 없었다.

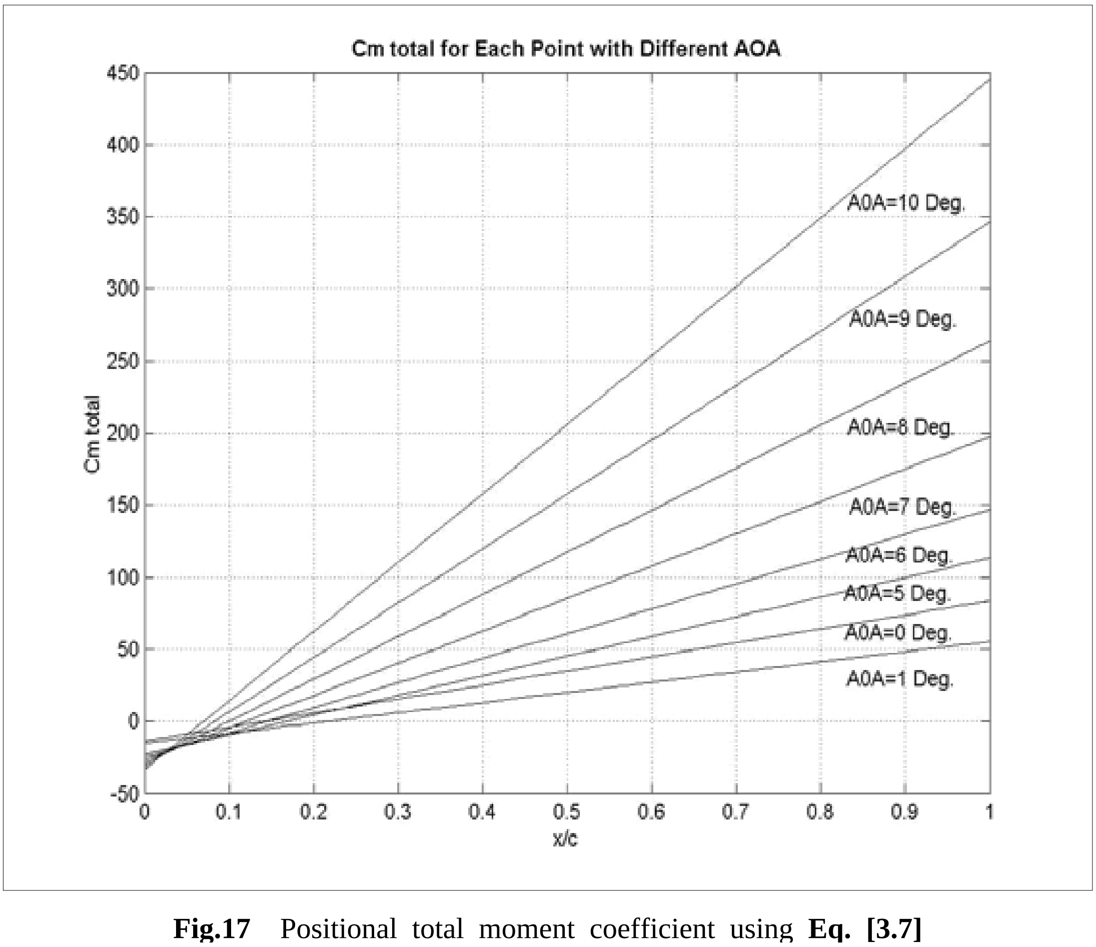

Fig.17 Positional total moment coefficient using **Eq. [3.7]**

반면 Ref.[4]의 식 [2.48]의 경우 정확성은 비교적 높으나 적용하는데 복잡성을 수반한다. 또한 본문에서 다루고자하는 Morphing 에어포일의 경우에는 mean chord line에 대한 적합 방정식을 쉽게 구할 수 없고, 계산도 용이하지 않기 때문에 다른 방법을 고찰하기로 한다. 이곳에서는 식 유도로만 제한하겠다.

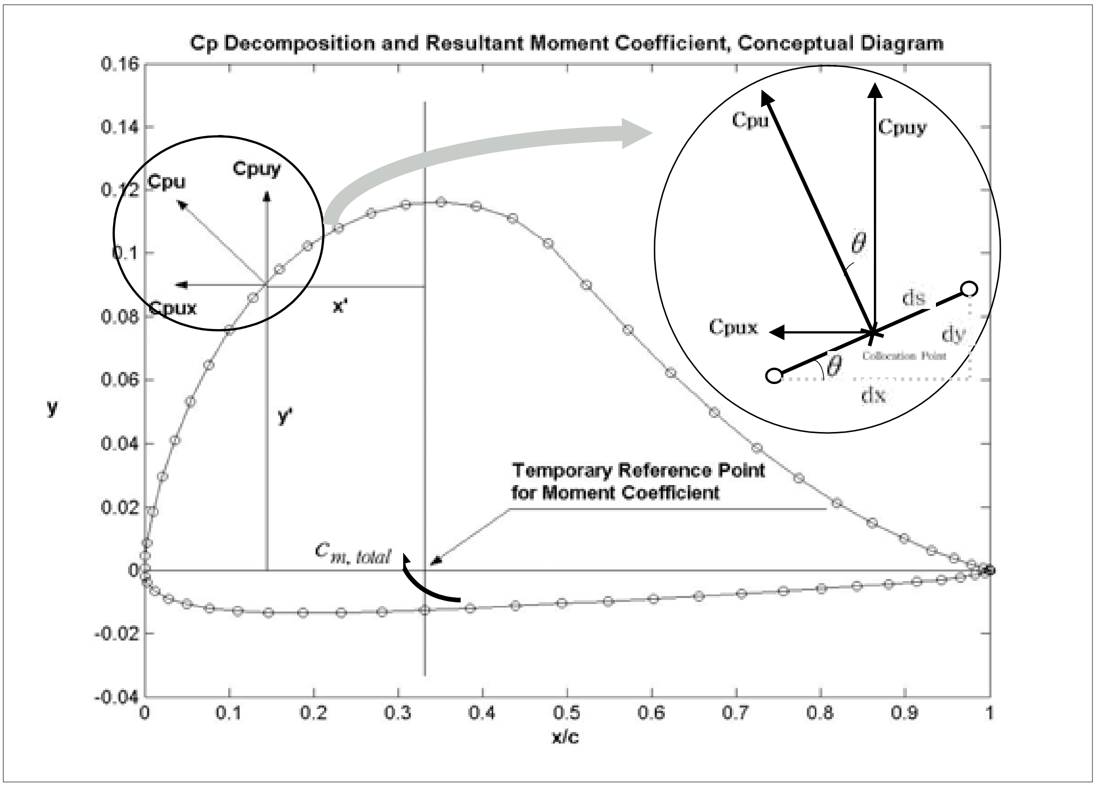

Fig.18 Cp decomposition and moment coefficient relation

Fig.18은 임의 기준점에 대한 Cp의 성분별 분해를 보여주며 그 관계는 식 [3.8] 및 식 [3.9]과 같다. upper와 lower는 부호를 가지므로 통합하면,

$$Cpy = Cp\ \cos\theta = Cp\left(-\frac{dx}{ds}\right) \quad [3.8]$$

$$Cpx = Cp\ \sin\theta = Cp\left(-\frac{dy}{ds}\right) \quad [3.9]$$

임의 지점(x_{pt})에서, 위치별 (총) 모멘트 계수($c_{m,\,@x_{pt}}$)는 평형방정식을 식 [3.10]과 같이 세워 얻을 수 있다.

$$
\begin{aligned}
c_{m,@x_{pt}} &= \sum_{x=0}^{1} c_m \\
&= \sum_{x=0}^{1} [(C_{py} \times x') + (C_{px} \times y')] \\
&= \sum_{x=0}^{1} [(C_p \cos\theta \times x') + (C_p \sin\theta \times y')] \\
&= \sum_{x=0}^{1} [(C_p \frac{dx}{ds} \times x') + (C_p \frac{dy}{ds} \times y')]
\end{aligned}
\qquad [3.10]
$$

식 [3.10]은 검증단계를 거치지 않았으나 식 [3.7] 보다는 더 구체적이고 식 [2.48]보다는 비교적 덜 복잡하다. 따라서 비교적 정확한 모멘트 계수를 낼 수 있으며, 공력중심이 위치별 모멘트 계수 선도에서 각 받음각에 대한 위치별 모멘트 선의 교차점임을 고려하면 공력중심 또한 상대적으로 높은 정확성을 보일 수 있다. 힘의 정의에 있어서 작용점 정의는 필수적이므로, 이에 대한 검증이 필요하다.

이곳에서는 정성적으로만 고려한다면, 우선 Fig.17에서 캠버를 수반하는 비대칭 에어포일에서 C_{pux}값은 C_{plx} 보다 크므로, 그래프는 전반적으로 앞전에서 더 큰 (-)모멘트 계수 값을 불러올 것이다. 그러므로 위치별 (총) 모멘트 선도는 더 높은 기울기를 갖을 것이다. 이는 교차점을 그림보다 더 뒤로 후퇴시키는 효과가 있을 것이다. Miley-mix의 경우는 더 큰 캠버를 가지고 있으므로 위 성향은 더욱 심화되어 공력중심이 에어포일의 중앙쪽으로 이동하는 경향을 보일 것이다.

4. 결론(Conclusion)

플로우 차트를 따라서 스플라인 및 NURB 곡선을 적용하여 새로운 이산 Morphing 에어포일을 생성하였다. NURB 표현이 더 효율적이고 정확함을 확인할 수 있었다. NURB 적합의 결과로 주기적으로 작은 압력 물결이 존재하였으며 이는 패널 재구성에 의해 해결할 수 있었다. 위 압력 분포를 얻기 위해 해석방법으로 Linear Strength Vortex Panel법을 적용하였으며 이에 대한 결과를 XFOIL로 검증하였다. 그 결과 별다른 차이가 없었다. 초기 가정에서 '두 비행 가능한 에어포일의 중간 단계(Morphing 단계) 에어포일은 비행에 합당한 공력선도를 보일 것이다.'라는 가정은 Morphing의 중간단계를 제외하고 거의 일치하였다. 이 단계에 생성된 거대한 압력 discrepancy에 대해 더 구체적인 고려가 필요하다.

공력중심을 구하기 위해 식 [3.7]을 적용하여 교차점을 살펴보았으나 이론적인 공력중심을 얻을 수 없었다. 이는 x방향의 압력계수가 반영되지 않은 것에서 기인했다고 판단하여, 비교적 복잡한 Anderson의 모멘트 계수식 [2.48]과 McCormick에 의한 간단한 모멘트 계수 계산식 [3.7]을 비교하였고, 결과로 더 직관적인 방법을 고안하였다. 이에 의한 공력중심 추세에 대해 정성적으로 고찰을 더하였다.

결론적으로 이곳에서 제시된 Morphing 개념과 방법은 타당하다고 판단되며, 다만 초기 이상화 가정사항에 더 실제적인 사항 첨가, 현실적인 Morphing wing의 구현 방안과 해당 날개에 대한 실제적 실험에 대해 추가적인 검증이 필요하다.

5. 참고문헌(Reference)

[1] Prock, C. Brian외 2인, 2002, “Morphing Airfoil Shape Change Optimization with Minimum Actuator Engery as an Objective", AIAA-2002-5401

[2] Gano, E. Shawn. and Renanud, E. John, 2002, “Optimized Unmanned Aerial Vehicle with Wing Morphing for Extended Range and Endurance", AIAA-2002-5668

[3] Cadogan, David 외 4인, 2003, “Inflatable and Rigidizable Wing Components for Unmanned Arial Vehicles", AIAA-2003-1801

[4] Anderson, D. John, 2001, Fundamentals of Aerodynamics, 3rd Ed., McGraw-Hill

[5] http://www.aae.uiuc.edu/m-selig/ads/coord_database.html

[6] Katz, Joseph and Plotkin, Allen, 2001, Low-Speed Aerodynamics, 2nd Ed., Cambridge Press

[7] Abbott, H. Ira and Von Doenhoff, E. Albert, 1945, “Summary of Airfoil Data”, NACA Report No.824

[8] 이관수, 2002, 공학도를 위한 수치해석, 源和

[9] Farin, Gerald, 1999, NURBS from Projective Geometry to Practical Use, 2nd Ed., AK Peters

[10] McCormick W. Barnes, 1995, Aerodynamics Aeronautics and Flight Mechanics, 2nd Ed., John Wiley & Sons

```
C  ****************************************************************************
C  * PROGRAM FOURIER APPROXIMATION and GAUSS ELIMINATION
C  *                              LOWFOURIER.f
C  ****************************************************************************
C
C  ****************************************************************************
C  * 1)WHAT THIS PROGRAM DOES:
C  *                  1548                                          .
C  *   THIS PROGRAM READS ALL THE AIRFOIL DATA LISTED IN
C  *   http://www.aae.uiuc.edu/m-selig/ads/coord_database.html
C  *   FILE: coord_seligFmt.zip (1548 FILES).   FILELIST.TXT WAS MADE
C  *   USING COPY-AND-PASTE OF THE DIRECTORY.
C  *   naca64a204.dat HAS BEEN ADDED INTO THE DIRECTORY WITH ALL
C  *   FILES FROM coord_seligFmt.zip.
C  *
C  *   THIS PROGRAM ESTIMATES CURVES BY CALCULATING AIRFOIL
C  *   COORDINATE DATA AND RETURNS VALUES INTO 'LOWFOURIER.TXT'.
C  *
C  * 2)FUNCTIONS:
C  *   -MAIN FUNCTION
C  *   -SUBROUTINE
C  *      FOURIER: READ COORDINATE INTO ARRAY AND CALCULATE FOURIER
C  *      MATRIXSET: OBTAIN ELEMENT OF AUGUMENTED MATRIX Ax=B in 3*4
C  *      ORDER: MATRIX CALCULATION FOR GAUSS ELIMINATION
C  *      ELIM: MATRIX CALCULATION FOR GAUSS ELIMINATION
C  *      BACKSB: MATRIX CALCULATION FOR GAUSS ELIMINATION
C  *      ROOTAB: ROOT OF A1 AND B1 RETURNING C
C  *
C  *
C  *
C  * 4)REFERENCE:
C  *  "STRUCTURED FORTRAN 77 FOR ENGINEERS AND SCIENTIST 5TH ED."
C  *                             DELORES M. ETTER. ADDISON-WESLEY, 1997
C  *  "NUMERICAL METHODS FOR ENGINEERS(KOREAN)"
C  *                             LEE. KWAN-SOO. WON-WHA. 2002
C  ****************************************************************************
C
C  ----------------------------------------------------------------------------
C  * COMMON CONSTANTS
C  * I,I,L:    COUNTING INDEX. L: NUMBER OF COORDINATE X OR Y
C  * LMIN:     INDEX OF X HAVING THE MINIMUM MAGNITUDE
C  * N:        NUMBER OF EQUATION. IN HERE. N=3
C  * Z:        UNIT NUMBER FOR CONTROLLING FILE
C  * BLANK:    CONSTANT FOR CONTROLLING FILE STRING
C  * NAMECHAR:FILE NAME STRING FOR READING FILE SETS FROM A LIST
C  * FILETOP: FOR READING UNKNOWN LENGTH OF STRING IN A AIRFOIL
C  *          FILE IN THE FIRST LINE
C  * DROWS,DCOLS: NUMBER OF ROWS AND COLUMS
C  * X, Y:     COORDINATE ARRAY OF AIRFOILS LIMITED IN SIZE BY MAX 400
C  * A:        3 BY 4 MATRIX FOR CALCULATING FOURIER APPROXIMATION COEF.
C  * S:        FOURIER APPROXIMATION COEFFICIENTS RETURNED
C  ----------------------------------------------------------------------------

C  ----------------------------------------------------------------------------
C  * MAIN FUNCTION
C  *  THIS FUNCTION OPEN FILES. CALLS SUBROUTINES AND PERFORMS GAUSS
C  *  ELIMINATION.  A FILE 'LOWERFOURIER.TXT' IS CREATED.
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------

      INTEGER I. I. Z. L. LMIN. BLANK, DROWS, DCOLS, N, PIVOT
      REAL X(400). Y(400), A(3,4), S(4)
      LOGICAL ERROR
      CHARACTER*(25) NAMECHAR
      CHARACTER*(25) FILENAME

      OPEN(UNIT=2, FILE='FILELIST.TXT', STATUS='OLD')

      DO 300 I=1,1548
      READ(2,*) NAMECHAR
         BLANK = INDEX(NAMECHAR,' ')
         FILENAME = NAMECHAR(1:BLANK)

      Z = 5

      OPEN(UNIT=Z, FILE=NAMECHAR(1:BLANK), STATUS='OLD')

         CALL COORDINATE(Z,L,X,Y)
         L = L - 1
         CALL MINIMUM(X,L,LMIN)
         CALL MATRIXSET(X,Y,L,LMIN,A)

         N = 3
         PIVOT = 1
         ERROR = .FALSE.
         DROWS = 3
         DCOLS = 4

   10 IF (PIVOT.LT.N.AND..NOT.ERROR) THEN
          CALL ORDER(A,DROWS,DCOLS,N,PIVOT,ERROR)
           IF (.NOT.ERROR) THEN
            CALL ELIM(A,DROWS,DCOLS,N,PIVOT)
            PIVOT = PIVOT + 1
           END IF
         GO TO 10
         END IF

         IF (ERROR) THEN
            PRINT*, 'NO UNIQUE SOLUTION'
         ELSE
            CALL BACKSB(A,DROWS,DCOLS,N,S)
            CALL ROOTAB(S)
            OPEN(UNIT=20,FILE='LOWFOURIER.TXT',STATUS='NEW')
            PRINT 30, S(1),S(2),S(3),S(4),FILENAME
            WRITE(20,30) S(1),S(2),S(3),S(4),FILENAME
   30 FORMAT(1X,F9.6,1X,F9.6,1X,F9.6,1X,F9.6,2X,A)
         END IF

  300 CONTINUE
         END
C-----------------------------------------------------------------------------


C  ----------------------------------------------------------------------------
C  * SUBROUTINE COORDINATE
C  *  THIS ROUTINE READS  X,Y COORDINATES INTO X AND Y ARRAYS
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
         SUBROUTINE COORDINATE(Z,L,X,Y)

         INTEGER Z. I. L
         REAL X(400). Y(400)
         CHARACTER*(40) FILETOP

         READ(Z,*) FILETOP
          L = 1
   10 READ(Z,*,END=15) X(L), Y(L)
          L = L + 1
         GO TO 10
   15 RETURN
         END
C-----------------------------------------------------------------------------

C  ----------------------------------------------------------------------------
C  * SUBROUTINE MINIMUM
C  *  THIS ROUTINE RECEIVES X(L). L FROM THE MAIN FUNCTION.
C  *  THEN THE MINIMUM VALUE OF X(L) AND L ARE RETURNED TO MAIN.
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
         SUBROUTINE MINIMUM(X,L,LMIN)

         INTEGER I. L. LMIN
         REAL X(400). MIN

         MIN = X(1)

         DO 10 I=2. L
            IF (X(I).LT.MIN) THEN
               MIN = X(I)
               LMIN = I
            END IF
   10 CONTINUE

         RETURN
         END
C-----------------------------------------------------------------------------

C  ----------------------------------------------------------------------------
C  * SUBROUTINE MATRIXSET
C  *  CALCULATE COMPONENTS OF FOURIER APPROXIMATION MATRIX
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------

         SUBROUTINE MATRIXSET(X,Y,L,LMIN,A)

         INTEGER I. L. LMIN. D
         REAL X(400), Y(400), A(3,4), PI

         D = L - LMIN + 1
         PI=3.141592

         A(1,1) = 0
         A(1,2) = 0
         A(1,3) = 0
         A(2,1) = 0
         A(2,2) = 0
         A(2,3) = 0
         A(3,1) = 0
         A(3,2) = 0
         A(3,3) = 0
         A(1,4) = 0
         A(2,4) = 0
         A(3,4) = 0

         DO 10 I=LMIN. L
           A(1,2)=A(1,2)+COS(PI*X(I))
           A(1,3)=A(1,3)+SIN(PI*X(I))
           A(2,2)=A(2,2)+(COS(PI*X(I))**2)
           A(2,3)=A(2,3)+(COS(PI*X(I))*SIN(PI*X(I)))
           A(3,3)=A(3,3)+(SIN(PI*X(I))**2)
           A(1,4)=A(1,4)+Y(I)
           A(2,4)=A(2,4)+(Y(I)*COS(PI*X(I)))
           A(3,4)=A(3,4)+(Y(I)*SIN(PI*X(I)))
   10 CONTINUE

         A(1,1)=REAL(D)
         A(2,1)=A(1,2)
         A(3,1)=A(1,3)
         A(3,2)=A(2,3)

         RETURN
         END
C-----------------------------------------------------------------------------

C  ----------------------------------------------------------------------------
C  * SUBROUTINE ORDER
C  *  MATRIX CALCULATION
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
         SUBROUTINE ORDER(A,DROWS,DCOLS,N,PIVOT,ERROR)

         INTEGER DROWS. DCOLS. N. ROW. RMAX. PIVOT. K
         REAL A(DROWS,DCOLS). TEMP
         LOGICAL ERROR

         RMAX = PIVOT
         DO 10 ROW=PIVOT+1,N
            IF (ABS(A(ROW,PIVOT)).GT. ABS(A(RMAX,PIVOT))) RMAX = ROW
   10 CONTINUE

         IF (ABS(A(RMAX,PIVOT)).LT. 1.0E-05) THEN
             ERROR = .TRUE.
         ELSE
            IF (RMAX.NE.PIVOT) THEN
               DO 20 K=1,N+1
                  TEMP = A(RMAX,K)
                  A(RMAX,K) = A(PIVOT,K)
```

```
            A(PIVOT,K) = TEMP
   20       CONTINUE
         END IF
      END IF

      RETURN
      END
C-----------------------------------------------------------------------

C  ---------------------------------------------------------------------
C  * SUBROUTINE ELIM
C  *  GAUSS ELIMINATION TO OBTAIN COEFFICIENTS
C  ---------------------------------------------------------------------
C-----------------------------------------------------------------------
      SUBROUTINE ELIM(A,DROWS,DCOLS,N,PIVOT)

      INTEGER DROWS, DCOLS, N, COL, ROW, PIVOT
      REAL A(DROWS,DCOLS), FACTOR

      DO 10 ROW=PIVOT+1,N
         FACTOR = A(ROW,PIVOT)/A(PIVOT,PIVOT)
         A(ROW,PIVOT) = 0.0
         DO 5 COL=PIVOT+1,N+1
            A(ROW,COL) = A(ROW,COL) - A(PIVOT,COL)*FACTOR
    5    CONTINUE
   10 CONTINUE

      RETURN
      END
C-----------------------------------------------------------------------

C  ---------------------------------------------------------------------
C  * SUBROUTINE BACKSB
C  *  BACK SUBSTITUTION FOR GAUSS ELIMINATION
C  ---------------------------------------------------------------------
C-----------------------------------------------------------------------
      SUBROUTINE BACKSB(A,DROWS,DCOLS,N,S)

      INTEGER DROWS, DCOLS, N, ROW, COL
      REAL A(DROWS,DCOLS), S(DROWS)

      DO 20 ROW = N,1,-1
         DO 10 COL=N, ROW+1, -1
            A(ROW,N+1) = A(ROW,N+1) - S(COL)*A(ROW,COL)
   10    CONTINUE
         S(ROW) = A(ROW,N+1)/A(ROW,ROW)
   20 CONTINUE

      RETURN
      END
C-----------------------------------------------------------------------

C  ---------------------------------------------------------------------
C  * SUBROUTINE ROOTAB
C  *   C = ROOT(A^2 + B^2)
C  ---------------------------------------------------------------------
C-----------------------------------------------------------------------
      SUBROUTINE ROOTAB(S)

      REAL S(4), Y

      Y = S(2)*S(2) + S(3)*S(3)
      S(4)= SQRT(Y)

      RETURN
      END
C-----------------------------------------------------------------------

C PROGRAM END
```

```
C***********************************************************************
C*      COEFFICIENT SORTING PROGRAM(USING BUBBLE-SORTING ALGORITHM)
C*                             COEFFSORT.f
C* THIS PROGRAM SORTS COEFFICIENTS(LOWFOURIER.TXT) CREATED BY
C* LOWFOURIER.f AND RETURNS LOWFOURIERSORT.TXT. COEFFICIENTS ARE
C* STORED AS CF AND FILE AND FILE NAMES AS FNAME. AFTER SORTING,
C* COEF AND FOILNAMES ARE RETURNED IN ASCENDING ORDER
C*
C*
C***********************************************************************
      INTEGER I
      REAL CF(1548,4),COEF(1548,4)
      CHARACTER*(18) FNAME(1548)
      CHARACTER*(18) FOILNAME(1548)

      OPEN(UNIT=10,FILE='LOWFOURIER.TXT',STATUS='OLD')

      DO 50 I=1,1548
      READ(10,*) CF(I,1),CF(I,2),CF(I,3),CF(I,4),FNAME(I)
      PRINT 20, CF(I,1),CF(I,2),CF(I,3),CF(I,4),FNAME(I)
   20 FORMAT(1X,F9.6,1X,F9.6,1X,F9.6,1X,F9.6,2X,A)
   50 CONTINUE

      CALL COEFFSORT(FNAME,CF,FOILNAME,COEF)

      DO 100 I=1,1548
      OPEN(UNIT=12,FILE='LOWFOURIERSORT.TXT',STATUS='NEW')
      PRINT 80, COEF(I,1),COEF(I,2),COEF(I,3),COEF(I,4),FOILNAME(I)
      WRITE (12,80), COEF(I,1),COEF(I,2),COEF(I,3),COEF(I,4),
     *             FOILNAME(I)
   80 FORMAT(1X,F9.6,1X,F9.6,1X,F9.6,1X,F9.6,2X,A)
  100 CONTINUE
      END
C-----------------------------------------------------------------------
      SUBROUTINE COEFFSORT(FNAME,CF,FOILNAME,COEF)

      INTEGER I
      REAL CF(1548,4),COEF(1548,4)
      REAL TEMP1,TEMP2,TEMP3,TEMP4
      CHARACTER*(18) FNAME(1548)
      CHARACTER*(18) FOILNAME(1548)
      CHARACTER*(18) TEMPNAME(1548)
      LOGICAL SORTED

      DO 10 I=1,1548
         COEF(I,1)=CF(I,1)
         COEF(I,2)=CF(I,2)
         COEF(I,3)=CF(I,3)
         COEF(I,4)=CF(I,4)
         FOILNAME(I)=FNAME(I)
   10 CONTINUE

      SORTED = .FALSE.
      FIRST=1
      LAST=1547

   15 IF(.NOT.SORTED) THEN
        SORTED = .TRUE.
        DO 20 I=FIRST,LAST
           IF (COEF(I,4).GT.COEF(I+1,4)) THEN
               TEMP1=COEF(I,1)
               TEMP2=COEF(I,2)
               TEMP3=COEF(I,3)
               TEMP4=COEF(I,4)
               TEMPNAME(I)=FOILNAME(I)
               COEF(I,1)=COEF(I+1,1)
               COEF(I,2)=COEF(I+1,2)
               COEF(I,3)=COEF(I+1,3)
               COEF(I,4)=COEF(I+1,4)
               FOILNAME(I)=FOILNAME(I+1)
               COEF(I+1,1)=TEMP1
               COEF(I+1,2)=TEMP2
               COEF(I+1,3)=TEMP3
               COEF(I+1,4)=TEMP4
               FOILNAME(I+1)=TEMPNAME(I)
               SORTED = .FALSE.
           END IF
   20   CONTINUE
        LAST = LAST - 1
        GO TO 15
      END IF

      RETURN
      END

C-----------------------------------------------------------------------
C PROGRAM END
```

```
C **********************************************************************
C  *                    PROGRAM REVERTING SORT
C  *                         REVERTSORT.f
C  *********************************************************************
C  ****************************************************************
C  *  THIS PROGRAM REVERT CONTER-CLOCKWISE ORDER OF AIRFOIL DATA TO
C  *  CLOCKWISE ORDER WHILE COUNTING FROM TOP TO BOTTOM
C  *
C  *
C  *********************************************************************

C  ---------------------------------------------------------------------
C  * MAIN FUNCTION
C  ---------------------------------------------------------------------
C-----------------------------------------------------------------------
      INTEGER I, L
      REAL PT(300,2), NEWPT(300,2)
      CHARACTER*(25) FILENAME, FILENAMEMAKE

      PRINT*, 'FILE NAME, DATA TO REVERT?'
      PRINT*, 'CASE SENSITIVE, INPUT EX) stage1.dat'
      READ*, FILENAME
      PRINT*, 'REVERTED DATA FILENAME TO MAKE?'
      PRINT*, 'CASE SENSITIVE, INPUT EX) stage1rev.dat'
      READ*, FILENAMEMAKE

      OPEN(UNIT=2,FILE=FILENAME,STATUS='OLD')
      OPEN(UNIT=3,FILE=FILENAMEMAKE,STATUS='NEW')

      CALL COORDINATE(L,PT)
      L = L - 1
      DO I=1,L
         NEWPT(L-I+1,1)=PT(I,1)
         NEWPT(L-I+1,2)=PT(I,2)
      ENDDO

      DO I=1,L
      PRINT 20, NEWPT(I,1),NEWPT(I,2)
      WRITE(3,20) NEWPT(I,1),NEWPT(I,2)
   20 FORMAT(1X,F10.5,2X,F10.5)
      ENDDO
      END

C  ---------------------------------------------------------------------
C  * SUBROUTINE COORDINATE
C  ---------------------------------------------------------------------
C-----------------------------------------------------------------------
      SUBROUTINE COORDINATE(L,PT)

      INTEGER L
      REAL PT(300,2)

       L = 1
   10 READ(2,*,END=15) PT(L,1), PT(L,2)
       L = L + 1
      GO TO 10
   15 RETURN
      END
C-----------------------------------------------------------------------
C PROGRAM END
```

```fortran
C  ****************************************************************************
C  *          PROGRAM SPLINE FITTING AND REGENERATION OF AIRFOIL
C  *                                SPLINEFIT.f
C  *
C  *  SPLINE ROUTINE SOURCE: PROF. SA, JONGYUP
C  * ***************************************************************************
C
C  ----------------------------------------------------------------------------
C  * MAIN FUNCTION
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
      COMMON   M,N1,AI(4,300),DATAX(300),X(300),Y(300)

      INTEGER I, L, LMIN, LIMIT
      REAL PT(300,2), NEWPT(300,2), DX
      CHARACTER*(25) FILENAME, FILENAMEMAKE


      PRINT*,'FILE NAME FOR SPLINE AND FOR NEW COORDINATES,FULL FORMAT?'
      READ*, FILENAME
      PRINT*,'FILE NAME TO BE CREATED?'
      READ*, FILENAMEMAKE

      OPEN(UNIT=2, FILE=FILENAME, STATUS='OLD')
      OPEN(UNIT=3, FILE=FILENAMEMAKE, STATUS='NEW')

      CALL COORDINATE(L,PT)
      L = L - 1

      CALL MINIMUM(PT,L,LMIN)
      LIMIT = LMIN

      CALL SORT(LIMIT,PT,NEWPT)

      N1=LIMIT
      AI(2,1)=0
      AI(2,N1)=0
      PRINT*, 'DX?(0<DX<1)'
      READ*, DX

      DO I=1,N1
         DATAX(I)=NEWPT(I,1)
         AI(1,I)=NEWPT(I,2)
      ENDDO

      M1 = IFIX((DATAX(N1)-DATAX(1))/DX)
      M = M1 + 1
      DO 10 I =1,M+1
         X(I) = DATAX(1) + DX*(I-1)
   10 CONTINUE

      CALL SPLINE

      WRITE(6,200)
      WRITE(6,210) (I,X(I),Y(I),I=1,M+1)
      WRITE(3,220) (X(I),Y(I),I=1,M+1)
      STOP
  200 FORMAT(1H ,'SOLUTION ',/2X,'NO',8X,'POINT X',5X,'Y VALUE')
  210 FORMAT(1X,I3,3X,F10.5,7X,F10.5)
  220 FORMAT(1X,F10.4,2X,F10.4)


      END
C-----------------------------------------------------------------------------


C  ----------------------------------------------------------------------------
C  * SUBROUTINE COORDINATE
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
      SUBROUTINE COORDINATE(L,PT)

      INTEGER L
      REAL PT(300,2)
      CHARACTER*(40) FILETOP
      READ(2,*) FILETOP
       L = 1
   10 READ(2,*,END=15) PT(L,1), PT(L,2)
       L = L + 1
      GO TO 10
   15 RETURN
      END
C-----------------------------------------------------------------------------


C  ----------------------------------------------------------------------------
C  * SUBROUTINE MINIMUM
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
      SUBROUTINE MINIMUM(PT,L,LMIN)

      INTEGER I, L, LMIN
      REAL PT(300,2), MIN

      MIN = PT(1,1)

      DO 10 I=2, L
         IF (PT(I,1).LT.MIN) THEN
            MIN = PT(I,1)
            LMIN = I
         END IF
   10 CONTINUE

      RETURN
      END
C-----------------------------------------------------------------------------

C  ----------------------------------------------------------------------------
C  * SUBROUTINE SORT
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
      SUBROUTINE SORT(LIMIT,PT,NEWPT)

      INTEGER I, LIMIT, LAST, FIRST
      REAL PT(300,2), NEWPT(300,2)
      REAL TEMP1,TEMP2
      LOGICAL SORTED


      DO 10 I=1,LIMIT
         NEWPT(I,1)=PT(I,1)
         NEWPT(I,2)=PT(I,2)
   10 CONTINUE

      SORTED = .FALSE.
      FIRST=1
      LAST=LIMIT-1

   15 IF(.NOT.SORTED) THEN
        SORTED = .TRUE.
        DO 20 I=FIRST,LAST
           IF (NEWPT(I,1).GT.NEWPT(I+1,1)) THEN
               TEMP1=NEWPT(I,1)
               TEMP2=NEWPT(I,2)
               NEWPT(I,1)=NEWPT(I+1,1)
               NEWPT(I,2)=NEWPT(I+1,2)
               NEWPT(I+1,1)=TEMP1
               NEWPT(I+1,2)=TEMP2

               SORTED = .FALSE.
             END IF
   20   CONTINUE
        LAST = LAST - 1
        GO TO 15
      END IF

      RETURN
      END

C-----------------------------------------------------------------------------
C  ----------------------------------------------------------------------------
C  * SUBROUTINE SPLINE
C  ----------------------------------------------------------------------------
C-----------------------------------------------------------------------------

      SUBROUTINE SPLINE
      COMMON M,N1,AI(4,300),DATAX(300),X(300),Y(300)
      DIMENSION DELFI(300),DELXI(300),B(300)
      DATA B(1),DELXI(1)/1.0,0.01/

      N = N1 - 1

      DO 10 I = 2,N1
         DELXI(I) = DATAX(I) - DATAX(I-1)
         DELFI(I) = (AI(1,I)-AI(1,I-1))/DELXI(I)
   10 CONTINUE
      DO 20 I =2,N
         AI(2,I) = 3.0*(DELFI(I+1)*DELXI(I) + DELFI(I)*DELXI(I+1))
         B(I) = 2.0*(DELXI(I) + DELXI(I-1))
   20 CONTINUE
      DO 30 I =2,N
         G = DELXI(I+1)/B(I-1)
         B(I) = B(I) - G*DELXI(I-1)
         AI(2,I) = AI(2,I) - G*AI(2,I-1)
   30 CONTINUE
      DO 40 I =N,2,-1
         AI(2,I) = (AI(2,I) - DELXI(I)*AI(2,I+1))/B(I)
   40 CONTINUE
      DO 50 I =1,N
         F1 = DELFI(I+1)
         F2 = AI(2,I) + AI(2,I+1) - 2.0*F1
         F3 = DELXI(I+1)
         AI(3,I) = (F1 - AI(2,I) - F2)/F3
         AI(4,I) = F2/(F3**2)
   50 CONTINUE
      WRITE(6,200)
      DO 60 I =1,N1
         WRITE(6,210) I,AI(1,I),AI(2,I),AI(3,I),AI(4,I)
   60 CONTINUE
      DO 70 K =1,M+1
         I = 1
         IF(X(K).GE.DATAX(I)) GO TO 1000
         DO 80 J =I,1,-1
            IF(X(K).GE.DATAX(J)) GO TO 1100
   80    CONTINUE
 1000    DO 90 J =I,N
            IF(X(K) .LT. DATAX(J+1)) GO TO 1100
   90    CONTINUE
         J = N
 1100    I = J
         DIFX = X(K) - DATAX(I)
         Y(K) = AI(1,I) + DIFX*(AI(2,I) + DIFX*(AI(3,I) + DIFX*AI(4,I)))
   70 CONTINUE
      RETURN
  200 FORMAT(1H ,'THE COEFFICIENT OF SPLINE POLYNOMIAL',/
     *        2X,'NO.',10X,'A0',10X,'A1',10X,'A2',10X,'A3',/)
  210 FORMAT(1X,I3,2X,4F13.4)

      END

C-----------------------------------------------------------------------------
C PROGRAM END
```

```fortran
C  ********************************************************************************
C  *                    PROGRAM NURB CURVE FITTING
C  *                             NURB.f
C  *
C  *  NURB ROUTINE SOURCE: PROF. SA. JONGYUP
C  ********************************************************************************

C  --------------------------------------------------------------------------------
C  * MAIN FUNCTION
C  --------------------------------------------------------------------------------
C---------------------------------------------------------------------------------

      COMMON X(300), Y(300), T(500), CNIK(500), H(300)
      INTEGER I, L
      REAL KNOT

      OPEN(UNIT=2, FILE='naca64a204l.dat', STATUS='OLD')
      CALL COORDINATE(L,X,Y)

      DO I = 1, L
            H(I) = 1.
      ENDDO

      DO I = 1, 300
            CNIK(I) = 0.
      ENDDO
      DO I = 1, 300
            T(I) = 0.
      ENDDO

        N = L
        K = 2
        KNOT = 0.005

      DO I = 1, N+K+1
            IF((0.LE.(I-1)).AND.((I-1).LT.K)) THEN
                        T(I) = 0
            ELSE IF((K.LE.(I-1)).AND.((I-1).LE.N)) THEN
                        T(I) = (I-1) - K + 1
            ELSE IF((N.LT.(I-1)).AND.((I-1).LE.N+K)) THEN
                        T(I) = N - K + 2
            ENDIF
      ENDDO

      OPEN(UNIT=10, FILE='naca64a204lf.dat', STATUS='UNKNOWN')
      DO U = 0, N-K+2, KNOT
            TX = 0.
            TY = 0.
            HINIK = 0.
      CALL CALCNIK(N, K, U, L)
      DO I = 1, N+1
         TX = TX + CNIK(I) * X(I)
         TY = TY + CNIK(I) * Y(I)
         HINIK = HINIK + H(I) * CNIK(I)
      ENDDO
         TX = TX / HINIK
         TY = TY / HINIK
      WRITE(10, 200) TX, TY
  200 FORMAT(1X,F7.4,2X,F9.6)
      ENDDO
      CLOSE(10, STATUS='KEEP')

      STOP
      END

C---------------------------------------------------------------------------------
C  --------------------------------------------------------------------------------
C  * SUBROUTINE COORDINATE
C  *  THIS ROUTINE READS  X,Y COORDINATES INTO X AND Y ARRAYS
C  --------------------------------------------------------------------------------
C---------------------------------------------------------------------------------
      SUBROUTINE COORDINATE(L,X,Y)

      INTEGER I, L
      REAL X(300), Y(300)
      CHARACTER*(40) FILETOP
      READ(2,*) FILETOP
       L = 1
   10 READ(2,*,END=15) X(L), Y(L)
       L = L + 1
      GO TO 10
   15 RETURN
      END

C---------------------------------------------------------------------------------

    SUBROUTINE CALCNIK(N, K, U, L)
    COMMON X(300), Y(300), T(500), CNIK(500), H(300)
    INTEGER L
    DIMENSION TEMPN(500)

    DO IK = 2, K
    DO I = 1, N+K-1
    IF(IK.EQ.2) THEN
            IF(T(I).EQ.U) THEN
                 TEMPNIK = 1
            ELSE IF((T(I).LE.U).AND.(U.LT.T(I+1))) THEN
                 TEMPNIK = 1
            ELSE
                 TEMPNIK = 0
            ENDIF
            IF(T(I+1).EQ.U) THEN
                 TEMPNI1K = 1
            ELSE IF((T(I+1).LE.U).AND.(U.LT.T(I+2))) THEN
                 TEMPNI1K = 1
            ELSE
                 TEMPNI1K = 0
            ENDIF
        ELSE
        TEMPNIK = CNIK(I)
        TEMPNI1K = CNIK(I+1)
    ENDIF

    IF((T(I+IK-1) - T(I)).EQ.0) THEN
      TEMP1 = 0
    ELSE
      TEMP1 = ((U-T(I))*TEMPNIK)/(T(I+IK-1)-T(I))
    ENDIF

    IF((T(I+IK) - T(I+1)).EQ.0) THEN
      TEMP2 = 0
    ELSE
      TEMP2 = ((T(I+IK)-U)*TEMPNI1K)/(T(I+IK)-T(I+1))
    ENDIF

    TEMPN(I) = TEMP1 + TEMP2
    ENDDO

    DO I = 1, L
     CNIK(I) = TEMPN(I)
    ENDDO
    ENDDO
    RETURN
    END
C---------------------------------------------------------------------------------
C PROGRAM END
```

```fortran
C  ********************************************************************************
C  *                    PROGRAM EQEXCLUDE
C  *                         EQEXCLUDE.f
C  *  USE THE PROGRAM AFTER NURB.f
C  *
C  ********************************************************************************

C  ********************************************************************************
C  *  MAIN FUNCTION: THIS PROGRAM EXCLUDE THE EQUAL X VALUE
C  *  THE PROGRAM READS NURB CREATED AIRFOIL DATA AND EXCLUDE
C  *  EQUAL X VALUES
C  ********************************************************************************
C*********************************************************************************

      INTEGER I, L
      REAL PT(80000,2)

      OPEN(UNIT=2, FILE='naca64a204lf.dat', STATUS='OLD')
      OPEN(UNIT=4, FILE='naca64a204lfs.dat', STATUS='UNKNOWN')

      CALL COORDINATE(L,PT)

      DO 200 I=1, L-1
        IF (PT(I,1).EQ.PT(I+1,1)) THEN
          GO TO 200
        ELSE
          WRITE(4, 100) PT(I,1), PT(I,2)
  100 FORMAT(1X,F8.5,2X,F9.6)
      ENDIF
  200 CONTINUE
              CLOSE(4, STATUS='KEEP')
      STOP
      END
C  --------------------------------------------------------------------------------
C  * SUBROUTINE COORDINATE
C  *  THIS ROUTINE READS  COORDINATES
C  --------------------------------------------------------------------------------
C---------------------------------------------------------------------------------
      SUBROUTINE COORDINATE(L,PT)

      INTEGER L
      REAL PT(80000,2)

       L = 1
   10 READ(2,*,END=15) PT(L,1), PT(L,2)
       L = L + 1
      GO TO 10
   15 RETURN
      END
C---------------------------------------------------------------------------------
```

```fortran
C  ******************************************************************************
C  *                         PROGRAM EQUALIZATION
C  *                            EQUALIZATION.f
C  *
C  ******************************************************************************

C  ******************************************************************************
C  *  USE THE PROGRAM AFTER EQEXCLUDE.f
C  *  MAIN FUNCTION: SELECT EQUAL X VALUES OF UPPER AND LOWER NURB
C  *  CREATED AIRFOIL DATA. THIS PROGRAM SAVES RESULTING
C  *  COMMON X VALUES INTO THEIR OWN NEWLY CREATED FILES
C  ******************************************************************************
C-------------------------------------------------------------------------------
C     MAIN FUNCTION
C-------------------------------------------------------------------------------
      INTEGER I,J,L,L1,L2,Z
      REAL PT(80000,2),PT1(80000,2),PT2(80000,2)

      OPEN(UNIT=2, FILE='naca64a204ufs.dat', STATUS='OLD')
      OPEN(UNIT=4, FILE='naca64a204lfs.dat', STATUS='OLD')
      OPEN(UNIT=6, FILE='naca64a204lfsc.dat', STATUS='NEW')
      OPEN(UNIT=8, FILE='naca64a204ufsc.dat', STATUS='NEW')

      Z=2
      CALL COORDINATE(Z,L,PT)
      L1=L
      DO I=1, L1
      PT1(I,1)=PT(I,1)
      PT1(I,2)=PT(I,2)
      ENDDO

      Z=4
      CALL COORDINATE(Z,L,PT)
      L2=L
      DO I=1, L2
      PT2(I,1)=PT(I,1)
      PT2(I,2)=PT(I,2)
      ENDDO

      DO 60 I=1, L1
         DO 50 J=1, L2
           IF (PT1(I,1).EQ.PT2(J,1)) THEN
             WRITE(6, 100) PT1(I,1), PT1(I,2)
           ELSE
             GO TO 50
           ENDIF
   50    CONTINUE
   60 CONTINUE

      DO 80 I=1, L2
         DO 70 J=1, L1
           IF (PT2(I,1).EQ.PT1(J,1)) THEN
             WRITE(6, 100) PT2(I,1), PT2(I,2)
           ELSE
             GO TO 70
           ENDIF
   70    CONTINUE
   80 CONTINUE
  100 FORMAT(1X,F8.5,2X,F9.6)
      STOP
      END
C  ------------------------------------------------------------------------------
C  * SUBROUTINE COORDINATE
C  *  THIS ROUTINE READS   COORDINATES
C  ------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      SUBROUTINE COORDINATE(Z,L,PT)

      INTEGER Z, L
      REAL PT(80000,2)
       L = 1
   10 READ(Z,*,END=15) PT(L,1), PT(L,2)
       L = L + 1
      GO TO 10
   15 RETURN
      END
C-------------------------------------------------------------------------------
C PROGRAM END
C*******************************************************************************
C  *          PROGRAM : DISCRETIZATION OF MORPHING AIRFOILS
C  *                          DISCRETEFOIL.f
C  *
C*******************************************************************************
C*******************************************************************************
C*    THE PROGRAM GENERATES DISCRETE 'MORPHING' AIRFOILS
C*******************************************************************************
C-------------------------------------------------------------------------------
C     MAIN FUNCTION
C-------------------------------------------------------------------------------

      INTEGER NUM, N, I, J, K
      REAL HIGHPT(300,2),LOWPT(300,2),UNIT(300),NEWPT(300,2)
      CHARACTER*(25) HIGHNAME, LOWNAME, DISCRETFOIL

      PRINT*, 'HIGH THICKNESS/CHORD AIRFOIL NAME(FULL FORMAT)?'
      PRINT*, 'CASE SENSITIVE. INPUT EX) NACA64a204uprev.dat'

      READ*, HIGHNAME
      PRINT*, 'LOW THICKNESS/CHORD AIRFOIL NAME(FULL FORMAT)?'
      PRINT*, 'CASE SENSITIVE. INPUT EX) NACA64a204uprev.dat'
      READ*, LOWNAME

      OPEN(UNIT=20, FILE=HIGHNAME,STATUS='OLD')
      OPEN(UNIT=30, FILE=LOWNAME,STATUS='OLD')

      READ(20,*) (HIGHPT(I,1),HIGHPT(I,2),I=1,121)
      READ(30,*) (LOWPT(I,1),LOWPT(I,2),I=1,121)

      PRINT*, 'HOW MANY EQAL DISTANCE FOR DISCRETE AIRFOILS?'
      READ*, N

      NUM=N-1

      PRINT 90, NUM
   90 FORMAT('NUMBER OF DISCRETE MORPHING AIRFOILS:',1X,I3)

      DO I=1,121
         NEWPT(I,2)=LOWPT(I,2)
      ENDDO

      DO I=1,NUM
        PRINT*, '       '
        PRINT*, 'NAMES OF DISCRETE AIRFOIL?(N NUMBERS, FULL FORM)'
        PRINT*, 'CASE SENSITIVE. INPUT EX) stage1.dat'
        PRINT 100, I
  100 FORMAT(1X,I3,3X)
      READ*, DISCRETFOIL
      OPEN(UNIT=I, FILE=DISCRETFOIL, STATUS='NEW')

        DO J=1,121
          UNIT(J)=(HIGHPT(J,2)-LOWPT(J,2))/N
          NEWPT(J,2)=NEWPT(J,2)+UNIT(J)

          WRITE(I,150) LOWPT(J,1),NEWPT(J,2)
  150 FORMAT(1X,F12.5,2X,F12.5)
        ENDDO


      ENDDO

      PRINT*, 'DISCRETIZATION DONE'

      END

C-------------------------------------------------------------------------------
C PROGRAM END


C*******************************************************************************
C  *          PROGRAM LINEAR STRENGTH V0RTEX PANEL METHOD
C  *                              LVSP.f
C  * REFERENCE:Joseph Katz, Allen Plotkin,"Low-Speed Aerodynamics 2nd Ed.", Cambridge
C*******************************************************************************

      REAL EP(400,2), EPT(400,2), PT1(400,2), PT2(400,2)
      REAL CO(400,2), A(400,400), B(400,400), G(400)
      REAL TH(400), DL(400)

      OPEN(8,FILE='CPLV.dat')
      OPEN(9,FILE='mileyrev.dat')


      WRITE(6,*) 'ENTER NUMBER OF PANELS'
      READ(5,*) M
      N=M+1

      WRITE(6,*) 'ENTER ANGLE OF ATTACK IN DEGREES'
      READ(5,*) ALPHA
      AL=ALPHA/57.2958


      DO I=1,M+1
           READ(9,*) EP(I,1), EP(I,2)
      END DO

      DO I=1,M
           PT1(I,1)=EP(I,1)
           PT2(I,1)=EP(I+1,1)
           PT1(I,2)=EP(I,2)
           PT2(I,2)=EP(I+1,2)
      END DO

      DO I=1,M
           DZ=PT2(I,2)-PT1(I,2)
           DX=PT2(I,1)-PT1(I,1)
           TH(I)=ATAN2(DZ,DX)
      END DO

      DO I=1,M
          CO(I,1)=(PT2(I,1)-PT1(I,1))/2+PT1(I,1)
          CO(I,2)=(PT2(I,2)-PT1(I,2))/2+PT1(I,2)
      END DO

      DO I=1,M
        DO J=1,M

        XT=CO(I,1)-PT1(J,1)
        ZT=CO(I,2)-PT1(J,2)
        X2T=PT2(J,1)-PT1(J,1)
        Z2T=PT2(J,2)-PT1(J,2)

        X=XT*COS(TH(J))+ZT*SIN(TH(J))
        Z=-XT*SIN(TH(J))+ZT*COS(TH(J))
        X2=X2T*COS(TH(J))+Z2T*SIN(TH(J))
        Z2=0

      IF(I.EQ.1) THEN
        DL(J)=X2
      END IF

      R1=SQRT(X**2+Z**2)
      R2=SQRT((X-X2)**2+Z**2)
      TH1=ATAN2(Z,X)
      TH2=ATAN2(Z,X-X2)

      IF(I.EQ.J) THEN
        U1L=-0.5*(X-X2)/(X2)
        U2L=0.5*(X)/(X2)
        W1L=-0.15916
        W2L=0.15916
      ELSE
```

```
      U1L=-(Z*LOG(R2/R1)+X*(TH2-TH1)-
     *        X2*(TH2-TH1))/(6.23819*X2)
      U2L=(Z*LOG(R2/R1)+X*(TH2-TH1))/(6.23819*X2)
      W1L=-((X2-Z*(TH2-TH1))-X*LOG(R1/R2)+
     *        X2*LOG(R1/R2))/(6.28319*X2)
      W2L=((X2-Z*(TH2-TH1))-X*LOG(R1/R2))/(6.28319*X2)
     END IF

     U1=U1L*COS(-TH(I))+W1L*SIN(-TH(I))
     U2=U2L*COS(-TH(I))+W2L*SIN(-TH(I))
     W1=-U1L*SIN(-TH(I))+W1L*COS(-TH(I))
     W2=-U2L*SIN(-TH(J))+W2L*COS(-TH(J))

     IF(J.EQ.1) THEN
      A(I,1)=-U1*SIN(TH(I))+W1*COS(TH(I))
      HOLDA=-U2*SIN(TH(I))+W2*COS(TH(I))
      B(I,1)=U1*COS(TH(I))+W1*SIN(TH(I))
      HOLDB=U2*COS(TH(I))+W2*SIN(TH(I))
     ELSE IF(J.EQ.M) THEN
      A(I,M)=-U1*SIN(TH(I))+W1*COS(TH(I))+HOLDA
      A(I,N)=-U2*SIN(TH(I))+W2*COS(TH(I))
      B(I,M)=U1*COS(TH(I))+W1*SIN(TH(I))+HOLDB
      B(I,N)=U2*COS(TH(I))+W2*SIN(TH(I))
     ELSE
      A(I,J)=-U1*SIN(TH(I))+W1*COS(TH(I))+HOLDA
      HOLDA=-U2*SIN(TH(I))+W2*COS(TH(I))
      B(I,J)=U1*COS(TH(I))+W1*SIN(TH(I))+HOLDB
      HOLDB=U2*COS(TH(I))+W2*SIN(TH(I))
     END IF

     END DO

     A(I,N+1)=COS(AL)*SIN(TH(I))-SIN(AL)*COS(TH(I))

     END DO


     A(N,1)=1
     A(N,N)=1

     IF(M.EQ.10) THEN
      DO I=1,11
       WRITE(6,10) A(I,1),A(I,2),A(I,3),A(I,4),A(I,5),A(I,6),
     *              A(I,7),A(I,8),A(I,9),A(I,10),A(I,11)
      END DO
     END IF

     N=N+1

     CALL MATRX(A,N,G)

 200 CONTINUE

     N=M+1
     CL=0

     DO I=1,M
       VEL=0
     DO J=1,N
       VEL=VEL+B(I,J)*G(J)
     END DO
       V=VEL+COS(AL)*COS(TH(I))+SIN(AL)*SIN(TH(I))
       CL=CL+V*DL(I)
       CP=1-V**2
       WRITE(8,11) CO(I,1),CP
     END DO

     WRITE(6,*) ' '
     WRITE(6,*) 'LIFT COEFFICIENT=', CL
     STOP
  10 FORMAT(/,F6.2,1X,F5.2,1X,F5.2,1X,F5.2,1X,F5.2,1X,
   *      F5.2,1X,F5.2,1X,F5.2,1X,F5.2,1X,F5.2,1X,F5.2)
  11 FORMAT(1X,F12.8,2X,F12.8)
     END
C-----------------------------------------------------------------------
     SUBROUTINE MATRX(A,N,G)
```

```
      REAL A(400,400), TEMP(400,400), G(400)
      DO I=1,N-1
       G(I)=0
      END DO

      DO I=1,N-1
    5 IF(ABS(A(I,I)).LT. 0.0000001) GO TO 9
        P=A(I,I)
        DO J=I,N
          A(I,J)=A(I,J)/P
        END DO

        DO K=I+1,N-1
          P2=A(K,I)
          DO L=I,N
            A(K,L)=A(K,L)-P2*A(I,L)
          END DO
        END DO
      END DO

      DO I=N-1,1,-1
       G(I)=A(I,N)
       DO J=I+1,N-1
        A(I,I)=0
        G(I)=G(I)-A(I,J)*G(J)
       END DO
      END DO

      RETURN

    9 IF(I.NE.N-1) THEN
       DO J=1,N
        TEMP(I,J)=A(I,J)
        A(I,J)=A(I+1,J)
        A(I+1,J)=TEMP(I,J)
       END DO
       GO TO 5
      ELSE
       GO TO 10
      END IF

   10 WRITE(6,*) 'NO SOLUTION'
      STOP
      END

C-----------------------------------------------------------------------
C PROGRAM END



C   ********************************************************************
C   *
C   *  THIS PROGRAM CALCULATE CENTER OF AIR PRESSURE
C   *  BASED ON Eq. [3.7]
C   *
C   ********************************************************************
C   --------------------------------------------------------------------
C   * MAIN FUNCTION
C   --------------------------------------------------------------------
C-----------------------------------------------------------------------

      INTEGER I,J,L,LMIN
      REAL PT(300,2), CM(300)
      CHARACTER*(25) FILENAME
      CHARACTER*(25) FILENAMEMAKE

      PRINT*,'PRESSURE DATA FILE TO READ?'
      PRINT*,'CASE SENSITIVE. INPUT EX) CPmiley.dat'
      READ*, FILENAME

      PRINT*,'NAME OF CMtot DATA FILE TO MAKE?'
      PRINT*,'CASE SENSITIVE. INPUT EX) CMmiley.dat'
      READ*, FILENAMEMAKE
```

```
      OPEN(UNIT=4,FILE=FILENAME,STATUS='OLD')
      OPEN(UNIT=6,FILE=FILENAMEMAKE,STATUS='NEW')

      CALL COORDINATE(L,PT)
      L = L - 1
      CALL MINIMUM(PT,L,LMIN)

      DO 30 I=1,L
        CM(I) = 0
        DO 20 J=1,L
          IF (PT(I,1).EQ.PT(J,1)) THEN
          GOTO 20
          ELSE IF (PT(J,1).GT.PT(I,1)) THEN
            IF (PT(J,2).GT. 0.) THEN
             CM(I)=CM(I)-(PT(J,1)-PT(I,1))*PT(J,2)
            ELSE
             CM(I)=CM(I)+(PT(J,1)-PT(I,1))*PT(J,2)
            ENDIF
          ELSE IF (PT(J,1).LT.PT(I,1)) THEN
            IF (PT(J,2).GE. 0.) THEN
             CM(I)=CM(I)+(PT(I,1)-PT(J,1))*PT(J,2)
            ELSE
             CM(I)=CM(I)-(PT(I,1)-PT(J,1))*PT(J,2)
            ENDIF
          ENDIF
   20   CONTINUE

      WRITE(6,25) PT(I,1), CM(I)
   25 FORMAT(1X,F10.5,2X,F10.5)
   30 CONTINUE

      PRINT*,'DONE'
      END

C   --------------------------------------------------------------------
C   * SUBROUTINE COORDINATE
C   --------------------------------------------------------------------
C-----------------------------------------------------------------------
      SUBROUTINE COORDINATE(L,PT)

      INTEGER L
      REAL PT(300,2)

      L = 1
   10 READ(4,*,END=15) PT(L,1), PT(L,2)
      L = L + 1
      GO TO 10
   15 RETURN
      END

C-----------------------------------------------------------------------

C   --------------------------------------------------------------------
C   * SUBROUTINE MINIMUM
C   --------------------------------------------------------------------
C-----------------------------------------------------------------------
      SUBROUTINE MINIMUM(PT,L,LMIN)

      REAL PT(300,2)
      INTEGER I, L, LMIN
      REAL MIN

      MIN = PT(1,1)

      DO I=2, L
         IF (PT(I,1).LT.MIN) THEN
            MIN = PT(I,1)
            LMIN = I
         END IF
      ENDDO

      RETURN
      END
C-----------------------------------------------------------------------
C PROGRAM END
```

a_0	a_1	b_1	c	Foil Name
−0.000422	−0.000420	−0.000186	0.000459	goe188.dat
−0.003577	0.000839	−0.000337	0.000904	goe587.dat
0.004688	0.001093	0.000254	0.001122	glennmartin2.dat
−0.007161	−0.001461	−0.000700	0.001619	raf15.dat
−0.002371	−0.001119	−0.001234	0.001666	raf6.dat
0.005283	0.001356	−0.001082	0.001735	coanda3.dat
−0.006933	−0.002029	0.000620	0.002121	goe573.dat
0.004572	0.001670	−0.001450	0.002211	n11.dat
−0.004539	−0.001640	−0.001552	0.002258	goe611.dat
−0.008149	−0.001968	0.001337	0.002379	goe500.dat
−0.001658	−0.001324	−0.001999	0.002398	goe05k.dat
−0.001880	−0.001638	−0.001932	0.002533	goe09k.dat
−0.002378	−0.002369	0.001032	0.002584	goe113.dat
−0.008050	0.001254	0.002404	0.002711	m17.dat
−0.006891	−0.002754	−0.000813	0.002872	goe562.dat
−0.003323	−0.002364	−0.001644	0.002880	ag11.dat
0.002108	−0.000227	−0.002914	0.002923	fx2.dat
−0.010690	−0.002924	0.000882	0.003054	goe344.dat
0.004801	0.002049	−0.002358	0.003124	dga1138.dat
−0.007705	−0.002788	−0.001484	0.003158	goe572.dat
0.002422	−0.000433	−0.003325	0.003353	fx3.dat
−0.005216	−0.001859	0.002870	0.003419	goe316.dat
0.002560	0.000198	−0.003605	0.003610	fx1.dat
−0.007101	−0.003303	−0.001557	0.003651	usnps4.dat
−0.004957	−0.003052	−0.002012	0.003656	goe385.dat
−0.003412	−0.003383	−0.001500	0.003701	goe134.dat
−0.008240	−0.003671	0.000779	0.003752	usa27.dat
−0.010318	−0.003150	0.002262	0.003877	goe511.dat
−0.003850	−0.003704	−0.001190	0.003891	aquilasm.dat
−0.006289	−0.002226	0.003199	0.003898	m14.dat
−0.002936	−0.002518	−0.003036	0.003937	goe10k.dat
−0.008366	−0.003908	0.001062	0.004049	usa26.dat
−0.006574	−0.002592	0.003236	0.004146	goe167.dat
−0.005445	−0.002241	0.003768	0.004384	goe574.dat
−0.003904	−0.003935	−0.001941	0.004388	goe365.dat
−0.003971	0.004393	−0.000732	0.004454	n11h9.dat
−0.004947	−0.003651	0.002563	0.004460	ag09.dat
−0.007113	−0.000602	0.004479	0.004519	goe257.dat
−0.007383	−0.003565	0.002794	0.004530	goe321.dat
−0.010846	−0.003869	0.002385	0.004545	arad10.dat
−0.005445	−0.003473	0.003077	0.004640	ag10.dat
−0.003108	−0.003990	0.002491	0.004704	goe415.dat
−0.006825	0.000719	0.004670	0.004725	goe746.dat
−0.011640	0.000468	0.004724	0.004747	m21.dat
−0.003432	−0.002962	−0.004115	0.004901	goe06k.dat
−0.007268	0.001073	0.004802	0.004920	goe309.dat
0.001445	0.000257	0.004944	0.004951	eiffel428.dat
−0.021648	0.001879	−0.004755	0.005112	usa35b.dat
−0.006369	−0.002974	0.004223	0.005165	goe318.dat
−0.007148	−0.003878	0.003509	0.005230	goe286.dat
0.001354	0.000560	0.005292	0.005321	isa571.dat
−0.011549	0.003517	0.004165	0.005452	m24.dat
−0.004000	−0.002133	0.005048	0.005480	goe315.dat
−0.004725	−0.004812	−0.002658	0.005498	goe711.dat
−0.007688	−0.004735	0.002800	0.005501	goe324.dat
−0.023679	0.003624	−0.004279	0.005608	sc21006.dat
−0.007099	−0.004537	0.003391	0.005664	goe488.dat
−0.008286	−0.004239	0.003818	0.005705	goe322.dat
−0.006457	−0.005762	0.000252	0.005767	goe474.dat
−0.007023	−0.004585	0.003641	0.005855	goe323.dat
−0.004931	−0.005698	−0.001367	0.005859	ag08.dat
−0.001243	−0.004136	0.004189	0.005880	fx74080.dat
0.013873	0.004652	−0.003607	0.005886	eiffel385.dat
−0.000714	**−0.00350**	**−0.004881**	**0.006007**	**naca64a204.dat**
0.004304	0.000163	−0.006027	0.006029	fx4.dat
−0.009237	−0.002225	0.005649	0.006072	goe406.dat
−0.011324	−0.005694	−0.002818	0.006353	goe530.dat
−0.004134	−0.004367	−0.004625	0.006361	ag03.dat
−0.003440	−0.004531	−0.004505	0.006390	e593.dat
−0.006191	−0.006391	−0.000152	0.006393	goe595.dat
0.002720	0.001571	0.006201	0.006397	isa960.dat
−0.007813	−0.005744	0.002900	0.006435	goe490.dat
−0.005372	−0.005604	−0.003396	0.006553	goe613.dat
−0.003591	−0.006634	0.000916	0.006696	e174.dat
−0.003758	−0.006698	−0.000087	0.006699	e387.dat
−0.006616	−0.006488	0.001965	0.006779	goe586.dat
−0.003941	−0.005280	−0.004293	0.006805	ag47ct02r.dat
−0.008036	−0.005351	0.004245	0.006830	goe55.dat
−0.005736	−0.001219	0.006732	0.006841	goe100.dat
−0.006348	−0.006233	−0.002859	0.006858	spicasm.dat
−0.002337	−0.002593	−0.006421	0.006925	sc20402.dat
−0.003937	−0.006868	−0.001062	0.006950	sd6080.dat
−0.004959	0.000453	0.007011	0.007025	goe491.dat
−0.003539	−0.004380	−0.005525	0.007050	e598.dat
−0.009559	−0.006871	−0.001812	0.007106	goe503.dat
−0.006010	−0.006619	0.002700	0.007149	goe591.dat
−0.005030	−0.003860	−0.006061	0.007186	goe07k.dat
−0.006948	0.000043	−0.007228	0.007228	m4.dat
−0.004679	−0.007236	−0.001477	0.007385	s9037.dat
0.004220	0.001503	0.007260	0.007414	coanda1.dat
−0.004491	−0.005249	−0.005349	0.007494	uag8814320.dat
−0.004344	−0.006614	−0.003676	0.007566	e176.dat
−0.003942	−0.006443	−0.004034	0.007602	s3025.dat
−0.003653	−0.007585	0.000826	0.007630	dae51.dat
−0.002495	−0.006637	0.003767	0.007632	goe392.dat
0.000947	−0.007420	−0.001880	0.007654	ho358.dat
−0.004150	−0.007455	0.001794	0.007668	ag13.dat
−0.003121	−0.006996	0.007612	0.007677	goe54.dat
−0.001667	−0.007138	−0.002830	0.007679	nsu94097.dat
−0.006942	−0.007612	−0.001116	0.007694	goe389.dat
−0.005706	−0.004935	−0.005910	0.007699	goe11k.dat
0.001936	0.001630	0.007544	0.007718	isa961.dat
0.004171	0.001784	0.007631	0.007837	coanda2.dat
0.003896	0.002963	0.007284	0.007863	eiffel430.dat
−0.004170	−0.007222	0.003184	0.007893	ag18.dat
−0.003949	−0.006250	0.004851	0.007912	ag14.dat
−0.011435	−0.007605	−0.002305	0.007947	goe505.dat
−0.008390	−0.007133	−0.003547	0.007966	goe670.dat
−0.013336	−0.008041	−0.000297	0.008046	goe510.dat
−0.005993	−0.002649	−0.007615	0.008063	m10.dat
−0.004113	−0.005947	−0.005496	0.008098	ag46ct02r.dat
−0.012829	−0.003418	0.007343	0.008099	goe744.dat
−0.008195	−0.002018	0.007859	0.008114	goe407.dat
−0.008637	−0.008119	0.000022	0.008119	l7769.dat
0.006107	0.002877	−0.007632	0.008156	n9.dat
−0.004848	−0.006745	−0.004589	0.008158	ag04.dat
−0.004012	−0.008165	0.001516	0.008304	ag26.dat
−0.004756	−0.008350	−0.000061	0.008351	s4061.dat
−0.004304	−0.008311	−0.001036	0.008375	ag12.dat
−0.004134	−0.006249	−0.005593	0.008386	ag45ct02r.dat
0.006378	0.002984	−0.007850	0.008398	ag38.dat
−0.002889	−0.007813	−0.003243	0.008459	mh38.dat
0.004154	0.000938	0.008457	0.008509	stcyr24.dat
−0.005215	0.004928	−0.006988	0.008550	n6h10.dat
−0.005511	−0.005924	0.006242	0.008606	goe477.dat
−0.002536	−0.000191	0.008612	0.008614	goe391.dat
0.007853	0.002440	−0.008265	0.008618	daytonwright6.dat
−0.004434	−0.008698	−0.000799	0.008645	ag17.dat
0.005835	0.002783	−0.008199	0.008658	rhodesg30.dat
−0.003989	−0.006158	0.006115	0.008678	ag19.dat
−0.004281	−0.004849	−0.007205	0.008685	ht22.dat
−0.007775	−0.008442	0.002231	0.008731	goe116.dat
−0.007622	−0.007737	−0.004243	0.008824	wasp.sm.dat
−0.009304	−0.004265	−0.007745	0.008842	goe512.dat
−0.004362	−0.007318	−0.005072	0.008904	s4310.dat
0.003541	−0.001035	0.008860	0.008920	ma409.dat
−0.001977	−0.008142	−0.003744	0.008961	mh116.dat
−0.013629	−0.008733	0.002062	0.008974	usa29.dat
−0.006388	−0.003653	−0.008604	0.009003	fx77w121.dat
−0.004799	−0.008900	−0.001419	0.009012	s4110.dat
0.010623	0.003100	−0.008470	0.009020	curtisc72.dat
−0.003478	−0.003186	−0.006480	0.009059	miley.dat
−0.006502	−0.000595	0.009066	0.009085	goe408.dat
−0.004231	−0.006651	−0.006298	0.009159	raf26.dat
−0.011522	−0.009217	−0.000594	0.009237	cr1.dat
−0.002982	−0.009027	−0.002211	0.009294	e64.dat
−0.004798	−0.008477	−0.003911	0.009336	sa7038.dat
−0.007282	−0.007576	−0.005517	0.009372	goe507.dat
−0.004925	−0.009314	−0.001312	0.009406	usa46.dat
−0.004813	−0.009458	−0.000427	0.009467	goe602m.dat
−0.007175	−0.009443	0.000740	0.009472	goe682.dat
−0.005853	−0.009129	0.002610	0.009495	s4062.dat
−0.006399	−0.009406	0.001312	0.009497	s2091.dat
−0.004335	−0.006826	0.006693	0.009560	s4083.dat
−0.005683	−0.003862	0.008777	0.009589	goe137.dat
−0.004143	−0.006722	0.006909	0.009639	ag27.dat
−0.004638	−0.009368	−0.002484	0.009692	sd7032.dat
−0.004352	−0.006736	−0.006974	0.009696	ag45ct02r.dat
−0.006520	−0.009660	−0.001146	0.009728	sd7034.dat
0.006558	0.003129	−0.009212	0.009729	rhodesg32.dat
−0.006800	−0.005199	−0.008233	0.009737	goe08k.dat
−0.008649	−0.008020	−0.005596	0.009779	goe515.dat
−0.003484	−0.007145	0.006733	0.009817	s4320.dat
−0.005109	−0.003029	0.009344	0.009823	goe277.dat
−0.004665	−0.009599	−0.002171	0.009841	e392.dat
−0.004201	−0.000918	0.009829	0.009872	goe377.dat
−0.004658	−0.006877	−0.007132	0.009908	s3024.dat
−0.007725	−0.009896	−0.000499	0.009909	goe615.dat
−0.007579	−0.009190	−0.003728	0.009917	goe584.dat
−0.005036	−0.009938	0.000523	0.009952	sd7043.dat
−0.009100	0.003434	0.009365	0.009975	chen.dat
−0.004001	−0.006441	0.007711	0.010047	goe492.dat
−0.003863	0.000650	0.010129	0.010150	goe280.dat
−0.009776	−0.009358	−0.003933	0.010151	goe412.dat
−0.008399	−0.008977	−0.004827	0.010192	goe414.dat
0.008555	0.003568	−0.009553	0.010198	n22.dat
−0.003798	−0.009435	−0.003929	0.010220	ag25.dat
−0.002027	0.001395	0.010136	0.010232	goe187.dat
−0.003389	−0.003490	0.009667	0.010278	goe133.dat
0.007805	0.003654	−0.009617	0.010288	ag37.dat
−0.006475	−0.007525	−0.007044	0.010307	goe561.dat
0.000394	−0.007738	−0.006814	0.010310	ho308.dat
−0.004080	0.002634	0.009970	0.010312	goe590.dat
−0.003748	−0.003804	−0.009589	0.010316	sc20503.dat
−0.004575	−0.001871	0.010160	0.010331	goe122.dat
−0.006664	−0.006314	−0.008226	0.010370	goe795.dat
0.001366	0.001484	0.010286	0.010393	isa962.dat
−0.004404	−0.006526	−0.008112	0.010411	s4053.dat
−0.004946	−0.000097	0.010439	0.010440	m13.dat
−0.010300	−0.005848	−0.008651	0.010442	usa27m2.dat
−0.004974	−0.006500	−0.008207	0.010469	e178.dat
−0.011427	−0.010388	−0.001303	0.010470	goe404.dat
−0.043184	0.010281	0.002224	0.010519	e664ex.dat
−0.003548	−0.001014	0.010493	0.010542	goe274.dat
−0.003610	−0.009040	−0.005461	0.010561	sa6042.dat
−0.004485	−0.007019	−0.007903	0.010570	ag44ct02r.dat
−0.006096	−0.010409	−0.001969	0.010594	goe115.dat
−0.005845	−0.001896	0.010474	0.010644	goe287.dat
−0.005039	−0.005378	0.009248	0.010698	goe180.dat
0.000579	−0.008628	−0.006367	0.010723	ho359.dat
−0.004953	−0.007655	−0.007552	0.010753	e193.dat
0.008029	0.003854	−0.010061	0.010774	n10.dat
−0.005467	−0.009298	−0.005589	0.010848	goe565.dat
−0.007812	−0.008601	−0.006636	0.010863	goe438.dat
−0.007788	−0.008592	−0.006672	0.010878	goe435.dat
−0.006167	−0.005317	0.009500	0.010887	usa22.dat
−0.011006	−0.010774	−0.001707	0.010909	goe701.dat
−0.004208	−0.008268	−0.007131	0.010918	sd7037.dat
−0.004703	−0.010000	−0.004774	0.011081	ag16.dat
−0.005435	−0.011070	0.000816	0.011100	goe532.dat
−0.004598	−0.008189	−0.007519	0.011117	goe602.dat
−0.006278	−0.007533	0.008200	0.011135	goe117.dat
−0.004739	−0.009252	−0.006225	0.011151	s7075.dat
−0.007291	−0.004902	−0.010024	0.011159	goe518.dat
−0.005330	−0.006829	−0.008885	0.011206	goe622.dat
−0.006020	−0.004242	0.010448	0.011276	goe328.dat
0.007607	0.003629	−0.010687	0.011286	rhodesg34.dat
−0.011909	−0.011208	−0.001553	0.011315	mue139.dat
−0.005846	−0.006871	−0.009075	0.011383	s7055.dat
−0.006843	−0.010608	0.004246	0.011426	mh83.dat
−0.003008	−0.002506	0.011157	0.011435	goe101.dat
−0.008512	−0.007322	−0.008826	0.011468	goe12k.dat
−0.006979	−0.010756	−0.004065	0.011499	goe547.dat
−0.006881	0.003166	0.011058	0.011502	m16.dat
−0.012085	−0.011447	0.001152	0.011505	usa34.dat
0.001164	−0.009387	−0.070918	0.071537	e543.dat
−0.019557	−0.014252	−0.070113	0.071547	fx77w270.dat
0.001724	−0.009450	−0.070974	0.071601	e540.dat
−0.011595	−0.022513	−0.068440	0.072048	raf69.dat
−0.003176	−0.013759	−0.070990	0.072311	e477.dat
−0.002351	−0.022037	−0.069035	0.072467	naca633218.dat
−0.001684	−0.013527	−0.071665	0.072931	e520.dat
−0.002220	−0.020590	−0.070081	0.073043	naca643218.dat
−0.005377	0.004153	0.073459	0.073576	e378.dat
−0.011533	−0.001232	0.073775	0.073785	raf19.dat
−0.009076	−0.021946	−0.070864	0.074184	ls421mod.dat
−0.011185	−0.021294	−0.071296	0.074408	fx77w258.dat
−0.007981	−0.022078	−0.071122	0.074470	ls421.dat
−0.014849	−0.016500	−0.073252	0.075087	ear21c.dat
−0.011152	−0.032770	−0.067601	0.075125	sc20518.dat
−0.012238	−0.020070	−0.072584	0.075308	us1000root.dat
−0.001810	−0.017717	−0.073462	0.075568	naca653218.dat
−0.009046	−0.020525	−0.072957	0.075789	naca633018.dat
−0.015355	−0.020607	−0.072985	0.075839	naca0018.dat
−0.001079	−0.025822	−0.071959	0.076452	naca634421.dat
−0.021721	0.000000	−0.076687	0.076687	dbln526.dat
−0.001045	−0.024196	−0.073032	0.076936	naca644421.dat
0.015751	−0.024568	−0.073145	0.077161	ua79sfm.dat
−0.008180	−0.014903	−0.076228	0.077671	s1014.dat
−0.003095	−0.013507	−0.076686	0.077867	naca663218.dat
−0.013787	−0.017996	−0.077615	0.079674	fx77w270s.dat
−0.000385	−0.020838	−0.077175	0.079939	naca654421.dat
−0.000601	0.010230	0.079378	0.080034	e376.dat
−0.003100	−0.014413	−0.078969	0.080274	e476.dat
−0.004594	0.004624	0.081391	0.081522	e377.dat
−0.009714	−0.016407	−0.080254	0.081914	s1016.dat
−0.018359	−0.026060	−0.078761	0.082960	naca2424.dat
−0.009665	−0.018627	−0.080872	0.082990	naca23021.dat
−0.013216	−0.005791	−0.082861	0.083063	naca16018.dat
−0.005828	−0.016322	−0.081895	0.083505	e838.dat
−0.011625	−0.018762	−0.082293	0.084405	b29root.dat
−0.017325	−0.022191	−0.081842	0.084797	fx6974.dat
−0.003517	0.010770	0.084784	0.085466	e377m.dat
−0.002537	−0.025889	−0.082193	0.086174	naca634221.dat
−0.004062	−0.015259	−0.084940	0.086299	tb25816.dat
−0.008378	0.003555	0.086444	0.086517	goe464.dat
−0.002500	−0.024260	−0.083279	0.086740	naca644221.dat
−0.017100	−0.024396	−0.083703	0.087186	goe775.dat
−0.004854	−0.012302	−0.086501	0.087371	naca66-018.dat
−0.017916	−0.024041	−0.085148	0.088477	naca0021.dat
−0.017360	−0.037317	−0.081194	0.089359	ah63w257.dat
−0.014165	−0.021857	−0.086901	0.089608	goe460.dat
−0.001833	−0.030887	−0.087450	0.089910	naca654221.dat
−0.014780	−0.028256	−0.085317	0.090824	raf89.dat
−0.003243	−0.016002	−0.091521	0.092910	naca664221.dat
−0.015418	−0.006755	−0.096673	0.096908	naca16021.dat
−0.021118	−0.012219	0.096890	0.097658	goe531.dat
−0.022136	−0.022968	−0.097628	0.100301	fx77w343.dat
−0.020472	−0.027476	−0.097318	0.101122	naca0024.dat
−0.028811	−0.036928	−0.095045	0.101967	ah63w300.dat
−0.004742	−0.016035	−0.101779	0.103035	n66021.dat
−0.020307	−0.029128	−0.099612	0.103783	goe776.dat
−0.027683	−0.041994	−0.102361	0.110548	ah94w301.dat
−0.013210	−0.000212	0.121664	0.121664	goe462.dat
−0.024972	−0.036956	−0.118663	0.124285	e862.dat
−2.741239	0.021027	0.123379	0.125158	n64215.dat
−0.089313	0.032846	−0.131964	0.135990	ah93w480b.dat
−0.026906	−0.038754	−0.134389	0.139865	e863.dat
−0.051906	−0.000224	−0.152682	0.152682	fx79w470a.dat
−0.028841	−0.040167	−0.148812	0.154137	e864.dat
−0.003579	−0.023554	−0.154293	0.156070	griffith30SS.dat
−0.085208	0.017609	−0.250915	0.251532	fx79w660a.dat

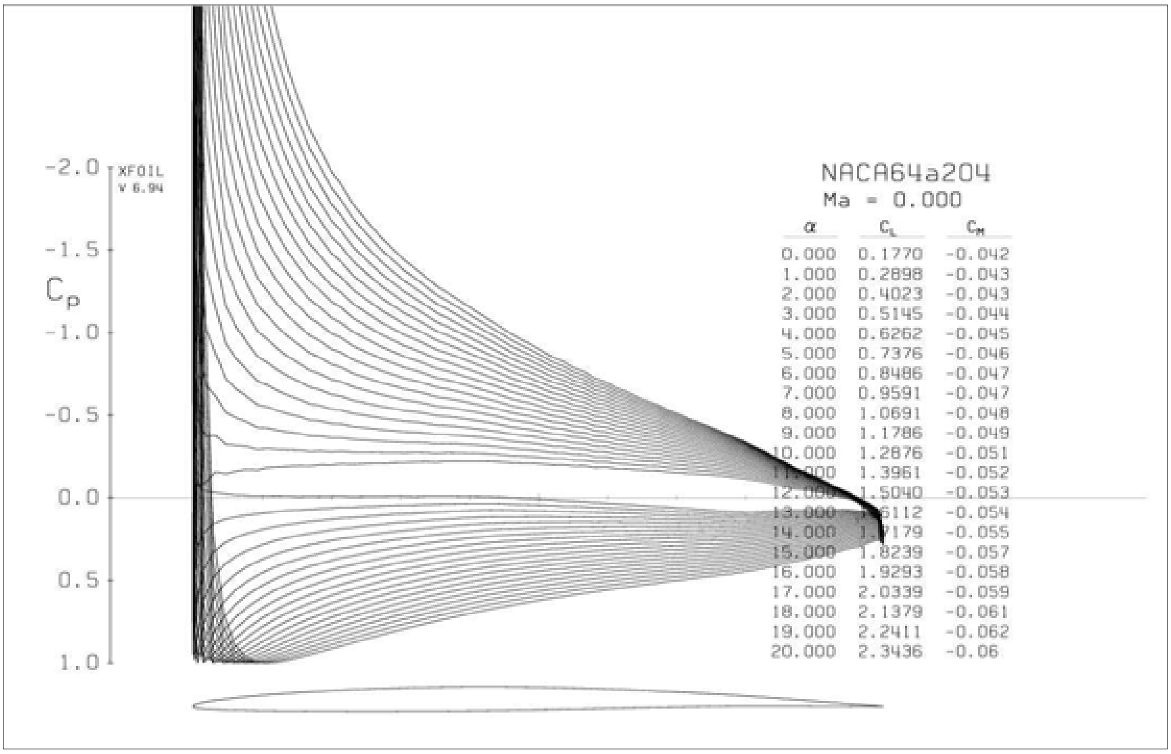

Figure Pressure Distribution trend for **NURB** generated NACA64a204

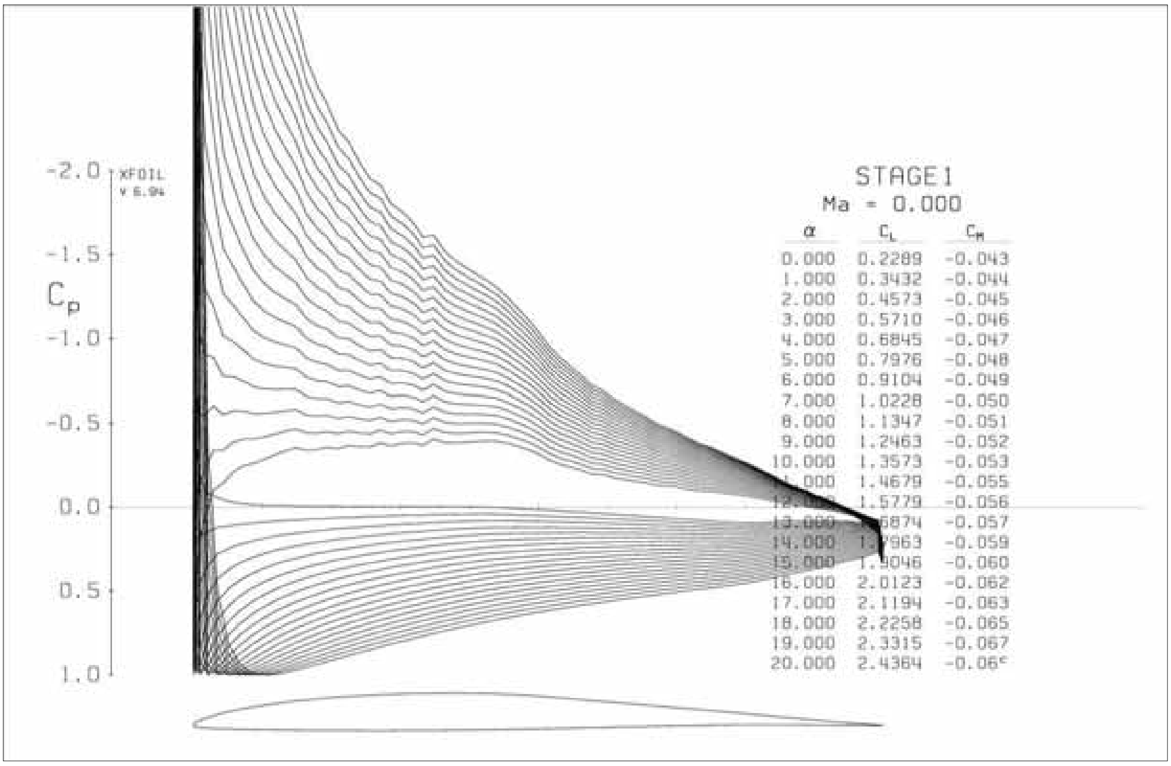

Figure Pressure Distribution trend for **NURB** generated Morphing Stage1 airfoil

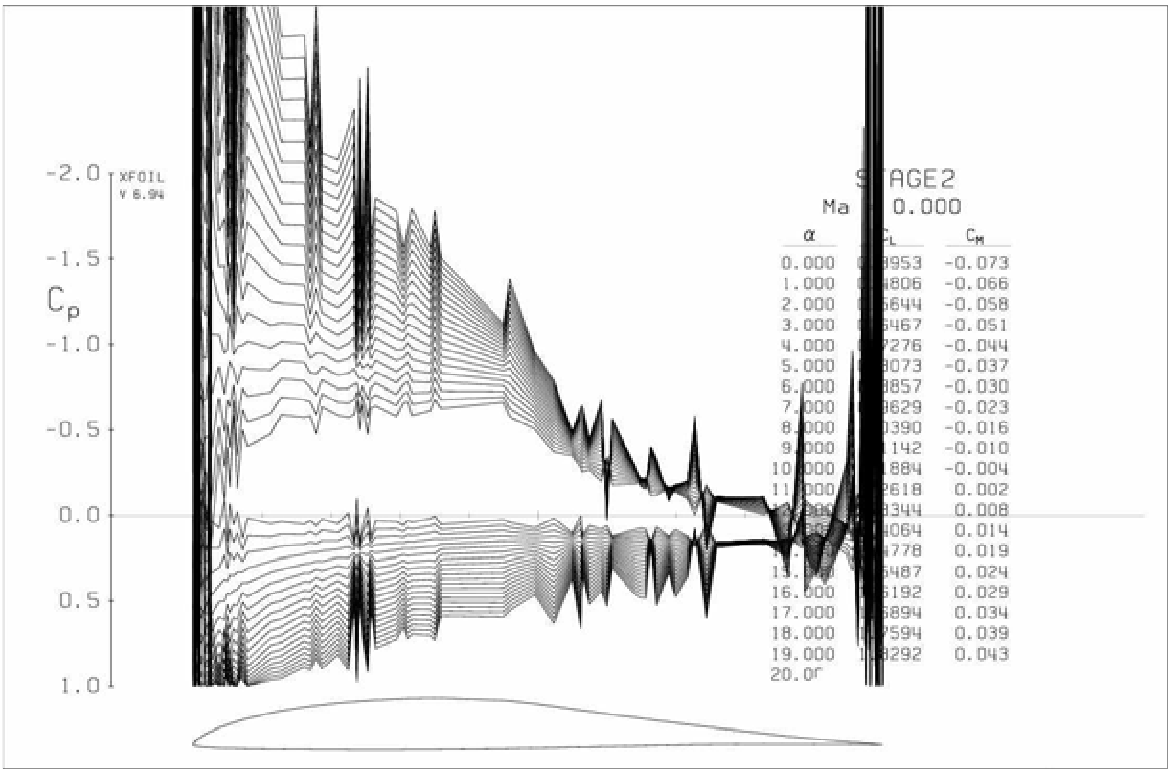

Figure Pressure Distribution trend for **NURB** generated Morphing Stage2 airfoil

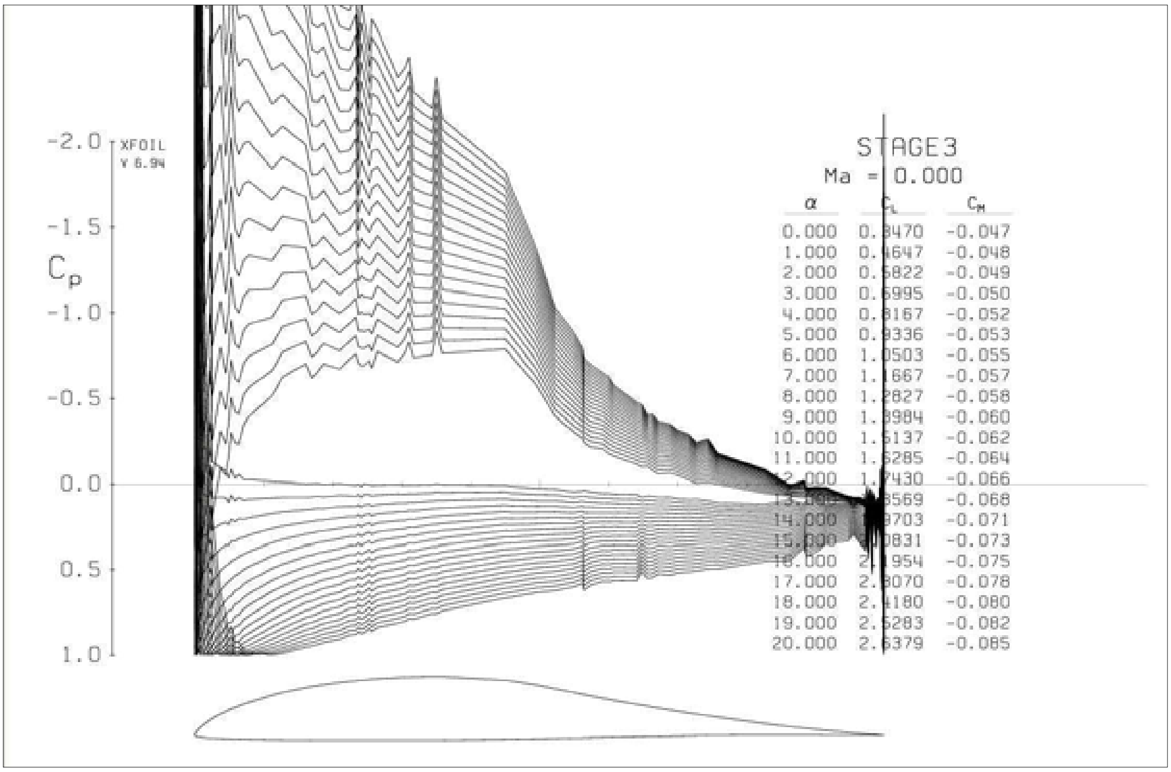

Figure Pressure Distribution trend for **NURB** generated Morphing Stage3 airfoil

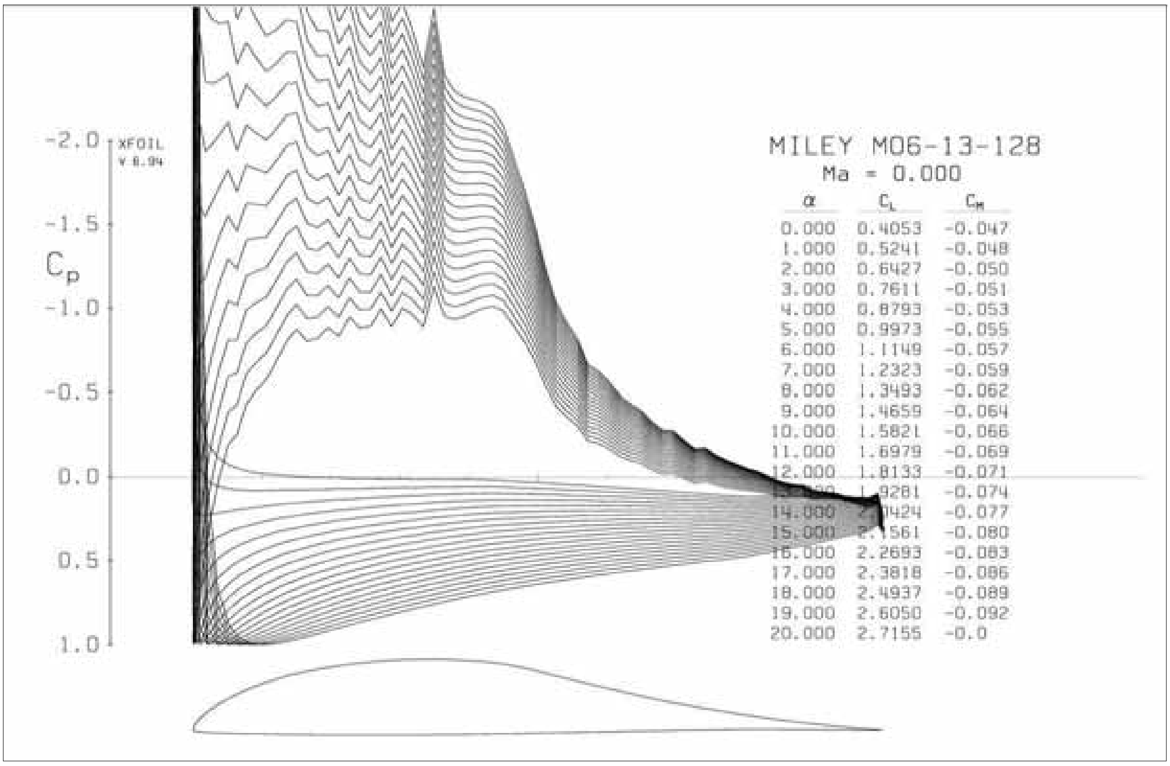

Figure Pressure Distribution trend for **NURB** generated Miley-mix airfoil

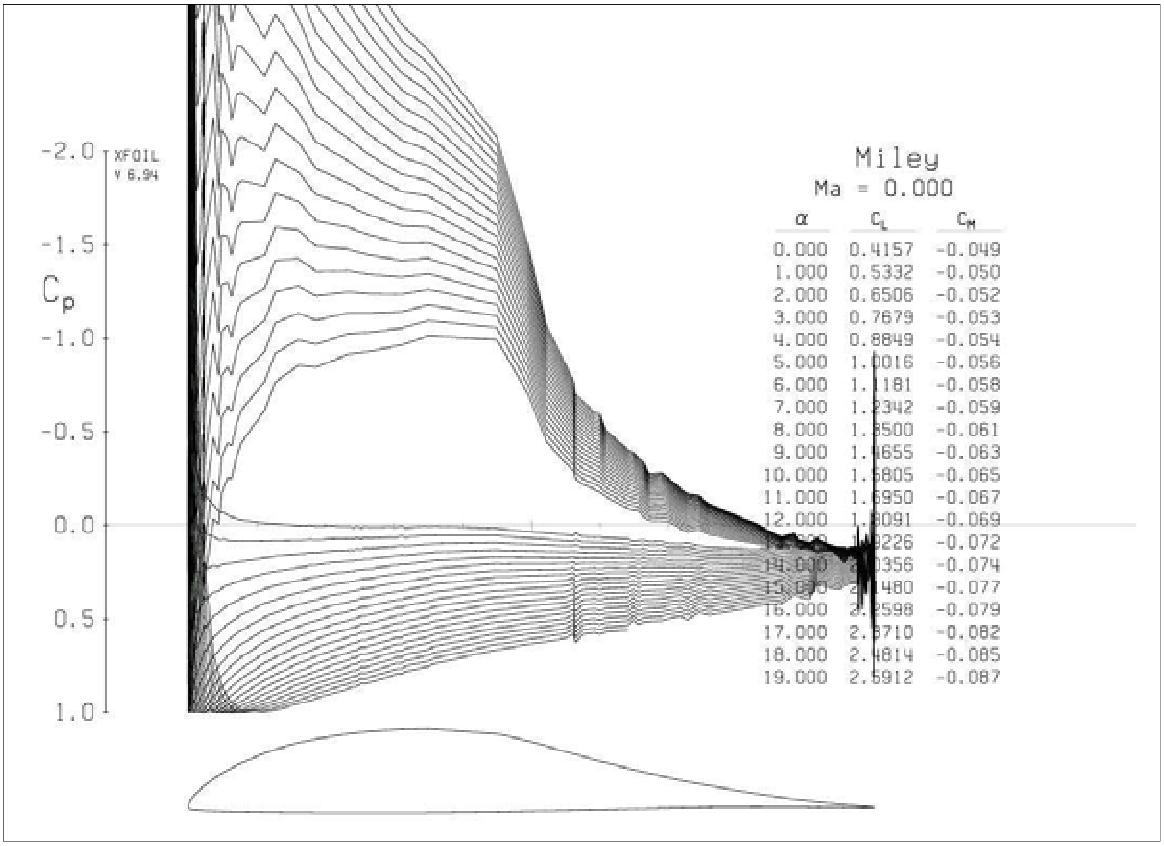

Figure Miley-mix panel reconfigured by piece