

## OTOMATİK TÜREV ARAÇLARI İLE AYRIK ADJOINT ÇÖZÜCÜ GELİŞTİRİLMESİ

Halil KAYA<sup>1</sup> ve Hakan TİFTİKÇİ<sup>2</sup>  
TUSAŞ, Ankara

İsmail Hakkı TUNCER<sup>3</sup>  
Orta Doğu Teknik Üniversitesi, ANKARA

### ÖZET

*Dış geometri aerodinamik eniyileme problemlerinde, gradyan tabanlı bir eniyileme yöntemi kullanıldığında amaç fonksiyonunun geometri kontrol parametrelerine göre hassasiyetinin hesaplanması gerekmektedir. Bu problemlerde dış geometriyi kontrol eden parametre sayısı yüzlerce, binlerce olabilmektedir ve sonlu farklar yöntemi ile ilgili hassasiyet değerlerinin hesaplanabilmesi için kontrol parametresi kadar akış çözümüne ihtiyaç duyulacaktır. Ancak adjoint yöntem kullanılarak, kontrol parametresi sayısından bağımsız olarak, bir akış çözüm süresi ile eş değer zamanda bir amaç fonksiyonunun tüm kontrol parametrelerine hassasiyeti hesaplanabilmektedir. İlgili hassasiyet değerlerini hesaplayabilen bir ayrık adjoint çözücü geliştirilmesi için, artık ve amaç fonksiyonlarının, çözüm ağı düğüm noktaları ve akış değişkenlerine göre kısmi türev değerlerini hesaplayan rutinlerin oluşturulması gerekmektedir. İlgili rutinlerin oluşturulması, doğrulanması zahmet verici bir süreç gerektirmektedir. Fakat kaynak kod değişimi mantığıyla çalışan otomatik türev (OT) araçları, sağlanan her türlü hesaplama rutinine zincir kuralını sistematik olarak uygulayarak, rutin çıkışı değişkenlerinin, girdi değişkenlerine göre türevini hesaplayabilen yeni bir rutini zahmetsizce ve otomatik olarak üretebilirler. Ayrıca yöntem ile hesaplanan türev değerleri, sonlu farklar yöntemi ile hesaplanan yaklaşık değerlerin aksine gerçek türev değerleri olup tek hata kaynağı bilgisayar aritmetiğidir. Bu çalışmada, dış geometri eniyileme çalışmalarında ihtiyaç duyulan hassasiyet değerlerinin hesaplanması için otomatik türev araçları ve ayrık adjoint [Giles, 2003] yöntemin kullanıldığı melez bir yaklaşım benimsenmiştir. Bu yaklaşım ile adjoint yöntemin çok sayıda değişkene göre hassasiyet değerlerini bir akış çözüm süresi ile eş değer zamanda hesaplama yeteneğinden ve otomatik türev araçlarının türev hesaplayan rutinleri zahmetsizce üretebilme yeteneklerinden faydalanılmıştır. Yaklaşım sonlu hacimler yöntemi kullanan iki boyutlu bir akış çözücü için adjoint çözücü geliştirilerek gösterilmiş ve sonuçları sonlu farklar yöntemi ile elde edilen sonuçlarla kıyaslanmıştır.*

### GİRİŞ

Havacılıkta dış geometri tasarımı, yüksek oranda, birçok alternatif tasarımı tünel testlerine göre nispeten hızlı değerlendirme imkânı sunan hesaplamalı akışkanlar mekaniği (HAD) analizlerine ve HAD analiz araçları kullanılarak gerçekleştirilen eniyileme çalışmalarına dayanmaktadır. HAD analiz araçları ile gradyan tabanlı tasarım eniyileme çalışması yapılacağına, amaç fonksiyonunun, geometriyi tanımlayan tüm kontrol parametrelerine göre hassasiyet değerlerinin elde edilmesi gerekmektedir. Ancak, tasarımda geometriyi kontrol parametre sayısı yüzler, binler mertebesinde olduğunda, sonlu farklar yöntemi ile hassasiyet değerlerinin hesaplanması uygulamada mümkün olamamaktadır. Adjoint yöntem teorisi gereği kontrol parametre sayısından bağımsız, bir akış çözüm süresi ile yakın mertebede bir zamanda, ihtiyaç duyulan hassasiyet değerlerini hesaplayabilmektedir. Adjoint yöntemin hassasiyet değerlerini hesaplamada üstünlüğü göz önünde bulundurulduğunda, tasarım süreçlerinde faydalanılan HAD analiz araçlarına adjoint kabiliyetin

<sup>1</sup> Uzman Mühendis, E-posta: hkaya@tai.com.tr

<sup>2</sup> Kıdemli Uzman Mühendis, E-posta: htiftikci@tai.com.tr

<sup>3</sup> Prof. Dr., Havacılık ve Uzay Müh. E-posta: ismail.h.tuncer@ae.metu.edu.tr

kazandırılmasının önemi aşikârdır. HAD analiz aracı olarak kullanılan bir akış çözücüyü, ayrık adjoint çözüm kabiliyetinin kazandırılması için, akış çözücünün artık değerlerini hesaplayan ayrık denklemlerin ve amaç fonksiyonu hesaplayan ayrık denklemlerin, çözüm ağı düğüm noktalarına ve akış değişkenlerine göre kısmi türevlerini hesaplayan rutinlerin geliştirilmesi gerekmektedir. Ancak, ayrık denklemlerin kısmi türevlerinin elde edilmesi, kısmi türev değerlerini hesaplayan rutinlerin oluşturulması ve doğrulanması zahmet verici bir süreç gerektirmektedir. Bu noktada, otomatik türev araçlarından faydalanılabilmektedir. Kaynak kod değişimi mantığıyla çalışan otomatik türev araçları, sağlanan her türlü hesaplama rutinine zincir kuralını sistematik olarak uygulayarak, rutinin çıktı değişkenlerinin, girdi değişkenlerine göre türevini hesaplayabilen yeni bir rutini zahmetsizce ve otomatik olarak üretebilmektedir.

Bu çalışmada, dış geometri eniyileme çalışmalarında ihtiyaç duyulan hassasiyet değerlerinin hesaplanması için otomatik türev araçları ve ayrık adjoint yöntemin kullanıldığı melez bir yaklaşım benimsenmiş ve kanat profili üzerinde ağdasız akış için bir uygulama sunulmuştur.

### YÖNTEM

Çalışmada iki boyutlu, hücre-merkezli, kapalı, düzensiz çözüm ağı ile sonlu hacimler yöntemi kullanan ağdasız bir akış çözücü için adjoint çözücü geliştirilmiştir.

#### Yönetici Denklemler

İki boyutlu Euler denklemleri integral formda takip edilen şekilde yazılabilirler.

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} d\Omega + \oint_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} ds = 0 \quad (1)$$

Burada, muhafazakâr değişkenler vektörü  $\mathbf{W}$ , ve akı  $\mathbf{F}$ , şu şekildedir.

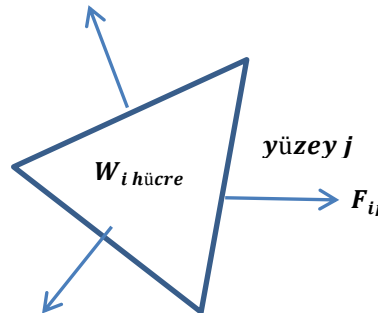
$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{F} = f\mathbf{i} + g\mathbf{j} \quad (2)$$

$$\mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho(E + p)u \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho(E + p)v \end{bmatrix} \quad (3)$$

Basınç değeri is ideal gaz denkleminde elde edilir.

$$p = (\gamma - 1)\rho \left[ E - \frac{u^2 + v^2}{2} \right] \quad (4)$$

Bu denklemler, serbest akım değerlerine ( $\rho_{\infty}, V_{\infty}, T_{\infty}$ ) ve referans uzunluk değerine (genelde veter uzunluğu) göre boyutsuzlaştırılır. Böylece boyutlu değerler benzer büyüklük değerlerinde incelenir ve yuvarlama kaynaklı nümerik hatalar asgari seviyeye indirilir. Sonlu hacimler yöntemi bir üçgen hücre için uygulanırsa,



Şekil 1 Üçgen bir hücre üzerinde sonlu hacimler gösterimi

Her yüzeydeki akım ayrı olarak hesaplanır.  $\oint_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} ds$  terimi ayrıık formda takip eden şekilde ifade edilir.

$$\oint_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} ds = \sum_{j=1}^3 \mathbf{F}_{ij} \cdot \mathbf{S}_{ij} \quad (5)$$

Bu durumda durağan, ağdasız akışlar için kısıt fonksiyonu her yüzeyden geçen akımın toplamı olacaktır.

$$\Omega \frac{\partial W}{\partial t} + \underbrace{\sum_{j=1}^3 \mathbf{F}_{ij} \cdot \mathbf{S}_{ij}}_{R_i} = 0 \quad (6)$$

### Adjoint Yöntem

Bir hava aracı etrafındaki akım için amaç fonksiyonunu ( $I$ ) tanımlayan aerodinamik özellikler, hava aracı geometrisini kontrol eden  $\mathbf{x}$  vektörüne ve akış alanı değişkenlerine ( $W$ ) bağlıdır. Bu durumda amaç fonksiyonu şu şekilde tanımlanabilir,

$$i = I(\mathbf{x}, W(\mathbf{x})) \quad (7)$$

Ayrıca akımı yöneten denklemler ve sınır koşulları, aerodinamik özelliklerin amaç fonksiyonu olduğu bir eniyileme probleminde kısıtlar olacaktır. Akımı yöneten denklemler ( $R$ ), akış alanı değişkenleri ( $W$ ) ile ilgili problemde kontrol değişkeni olan geometri ( $\mathbf{x}$ ) arasındaki dolaylı ilişkiyi vermektedir. Bu ilişki aşağıda verilen fonksiyon ile tanımlanabilir.

$$\mathbf{r} = \mathbf{R}(\mathbf{x}, W(\mathbf{x})) \quad (8)$$

Bu problemde amaç fonksiyonunun ( $I$ ) kontrol değişkeni olan geometriye ( $\mathbf{x}$ ) göre hassasiyet değerleri de aşağıdaki denklem ile bulunabilir.

$$\frac{di}{dx} = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial W} \frac{dW}{dx} \quad (9)$$

Denklem (9) ile amaç fonksiyonunun ( $I$ ), kontrol değişkenlerine göre ( $\mathbf{x}$ ) hassasiyeti elde edilmeye çalışıldığında, akış değişkenlerinin ( $W$ ), geometriye göre ( $\mathbf{x}$ ) hesaplanması ihtiyacı ortaya çıkacaktır. Ancak bu değer ( $dW/dx$ ) hesaplanması, diğer tüm kısmi türevlerin hesaplanmasına göre çok daha yüksek hesaplama ihtiyacı doğuracaktır. Çünkü geometride gerçekleşecek bir değişiklik, tüm akış hacmi değişkenlerinde değişikliğe yol açacaktır ve bu değişim akışın yeniden çözülmesi ile elde edilebilecektir.

Ancak akımı yöneten denklemler bir kısıttır ve her zaman geçerlidir. Bu ifadeyi denkleme dökerek olursak,

$$\frac{dr}{dx} = \frac{\partial R}{\partial x} + \frac{\partial R}{\partial W} \frac{dW}{dx} = 0 \quad (10)$$

Bu noktada ( $dW/dx$ ) değerinin hesaplanması için kısıt denklemlerinden faydalanabilecektir. Denklem (10) manipüle edilerek aşağıdaki şekilde ifade edilebilecektir.

$$\frac{dW}{dx} = - \left[ \frac{\partial R}{\partial W} \right]^{-1} \frac{\partial R}{\partial x} \quad (11)$$

Denklem (11), denklem (9) içinde kullanılırsa aşağıdaki şekilde bir ifade ile karşılaşılr.

$$\frac{dI}{dx} = \frac{\partial I}{\partial x} - \underbrace{\frac{\partial I}{\partial W} \left[ \frac{\partial R}{\partial W} \right]^{-1} \frac{\partial R}{\partial x}}_{\lambda} \quad (12)$$

Denklem (12) ile verilen sistemin, seçilecek yöntemle göre iki farklı çözümü olduğu görülmektedir. İlk yöntemde toplam türev değeri  $dW/dx$ , takip eden sistemin doğrudan çözülmesi ile hesaplanır. Bu yöntem direk yöntem olarak adlandırılmaktadır.

$$\frac{\partial R}{\partial W} \frac{dW}{dx} = - \frac{\partial R}{\partial x} \quad (13)$$

Bu yöntemin alternatifi ise adjoint yöntem olarak adlandırılmaktadır. Bu yöntemde  $\lambda$  olarak belirtilen bir matris tanımlanır. Bu yöntemde ise takip edilen sistemin çözülerek adjoint değişkenlerin bulunması gerekir.

$$\left[ \frac{\partial R}{\partial W} \right]^T \lambda = - \left[ \frac{\partial I}{\partial W} \right]^T \quad (14)$$

(13) ve (14) numaralı sistemlerden görülmektedir ki, direk yöntemde kontrol değişkenleri ( $x$ ) sayısı kadar, adjoint yöntemde ise amaç fonksiyon sayısı kadar denklem çözülmesi gerekmektedir. Aerodinamik dış geometri en iyileme problemlerinde genel olarak,  $CL, CD$  vb. birkaç değişken amaç fonksiyonu olarak karşımıza çıkarken, geometriyi kontrol eden parametre sayısı yüzlerce, binlerce olabilmektedir. Bu sebeple aerodinamik dış geometri en iyileme problemlerinde, adjoint yaklaşım çok yüksek fayda sağlamaktadır.

### Otomatik Türev

Otomatik türev (OT) [Griewank, 2000], bilgisayar programlarına zincir kuralının sistematik şekilde uygulamasını gerçekleştirmektedir. Otomatik türev, ayrıca algoritmik türev ve hesaplamalı türev isimleriyle de bilinmektedir.

Bilgisayar programları, ne kadar karmaşık olsalar dahi, her zaman temel aritmetik hesaplamalar ve fonksiyonlardan oluşmaktadırlar. Otomatik türev araçları da zincir kuralını bu hesaplamalar üzerinde uygulayarak herhangi bir programın çıktılarının girdilerine göre türevlerini hesaplayabilen programı üretirler. Dahası, yöntemin çıktıları analitik yöntem kadar doğrudur ve türev hesaplamaları otomatik olarak gerçekleştirileceğinden uygulamada yüksek kolaylık sağlamaktadırlar. Mevcut çalışmada otomatik türev aracı olarak Tapenade [Hascoet, 2004] programı kullanılmıştır.

Otomatik türev araçları, türev hesaplamaları için iki farklı yöntem sunarlar. Birinci yöntem, teğet ya da ileri mod türev, diğeri ise adjoint ya da ters mod türevdir. İleri mod türev, bir önceki bölümde bahsedilen direk yöntemle, ters mod türev ise adjoint yöntemle karşılık gelmektedir. İleri mod kullanıldığında, zincir kuralı içeriden dışarıya işlenir. Bir diğer deyişle, çıktının girdiye göre türevi hesaplanırken öncelikle çıktının girdiye olan ara bağılıklarının girdiye göre türevi hesaplanır. Bu sebepten her bir girdi için ayrı hesap yapılması gerekir. Ters modda ise zincir kuralı dışarıdan içeriye işlenir. Bir diğer deyişle, çıktının girdiye göre türevi hesaplanırken öncelikle çıktının girdiye olan ara bağılıklarının çıktıya göre türevi hesaplanır. Yöntemin matematiksel temeli kaynakçada [Martins, 2013] detaylı anlatılmıştır.

OT araçları ile otomatik türev hesaplamaları şu şekilde olacaktır. İfade (15)'de verilen değişken  $v$  vektörü, otomatik türev aracına verilen programda geçen her bir değişkeni ( $t$ ) içeren bir vektör olacaktır. Otomatik türev aracı, zincir kuralını programdaki her satır için uygulayacaktır. Bu durumda ilgili program bir seri açık fonksiyonlardan  $T_i$  oluşmuş şekilde değerlendirilecektir. Bu seride her fonksiyon sadece program girdilerine ya da serinin önceki adımlarında hesaplanmış fonksiyon değerlerine bağlı olacaktır. Böylece otomatik türev aracının aşağı ya da yukarı üçgen matrisleri çözmesi yeterli olacaktır.

$$v = \left[ \underbrace{t_1, \dots, t_{n_x}}_{\text{Bağımsız değişkenler}}, \underbrace{\dots, t_j, \dots, t_i}_{\text{Ara değişkenler}}, \underbrace{\dots, t_{n_f}, t_n}_{\text{Çıktı değişkenleri}} \right]^T \quad (15)$$

Değişkenler ve kısıtlar (her bir satırda tanımlanan eşitlikler) OT aracı tarafından şu şekilde algılanacaktır.

$$\mathbf{v} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}, \quad \mathbf{C}(\mathbf{v}) = \begin{bmatrix} t_1 - T_1() \\ t_2 - T_2(t_1) \\ \vdots \\ t_n - T_n(t_1, \dots, t_{n-1}) \end{bmatrix} \quad (16)$$

Ardından herhangi bir çıktı değişkeninin, herhangi bir girdi değerine göre türev değerleri Tablo 1 uyarınca yapılacaktır.

Tablo 1 İleri ve Ters Mod Türev Hesaplaması

İleri Mod	Ters Mod
$\frac{dt_i}{dt_j} = \delta_{ij} + \sum_{k=j}^{i-1} \frac{\partial T_i}{\partial t_k} \frac{dt_k}{dt_j}$	$\frac{dt_i}{dt_j} = \delta_{ij} + \sum_{k=j+1}^i \frac{dt_i}{dt_k} \frac{\partial T_k}{\partial t_j}$

Tablo 1 ile verilen ifade aşağıdaki şekilde matris formunda verilebilir.

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ -\frac{\partial T_2}{\partial t_1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -\frac{\partial T_n}{\partial t_1} & \dots & -\frac{\partial T_n}{\partial t_{n-1}} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ \frac{dt_2}{dt_1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \frac{dt_n}{dt_1} & \dots & \frac{dt_n}{dt_{n-1}} & 1 \end{bmatrix} = \mathbf{I} = \begin{bmatrix} 1 & -\frac{\partial T_2}{\partial t_1} & \dots & -\frac{\partial T_n}{\partial t_1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\frac{\partial T_n}{\partial t_{n-1}} \\ 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{dt_2}{dt_1} & \dots & \frac{dt_n}{dt_1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{dt_n}{dt_{n-1}} \\ 0 & \dots & 0 & 1 \end{bmatrix} \quad (17)$$

Yukarıdaki ifadede (17) sol taraf ileri moda, sağ taraf ise ters moda karşılık gelmektedir. Daha açık bir anlatımı mümkün kılmak için, otomatik türev uygulamasını bir Fortran rutini üzerinde göstermek yararlı olacaktır. Bu sebeple otomatik türev uygulaması Tablo 2' de verilen Fortran rutini üzerinde gösterilecektir.

Tablo 2 Örnek Fortran Rutini

Subroutine cost(x, y, c)	
real, intent(in)	:: x, y
real, intent(out)	:: c
real	:: z, k
z = 2*y + 5*x + 7	
k = z + y + x	
c = 3*x + y + z + k	
return	
end subroutine cost	

Tablo 2' de verilen "cost" isimli Fortran rutinde, (x,y) olarak adlandırılmış iki bağımsız değişken bulunmaktadır. Bu değişkenler rutinin girdi değerleridir. (z,k) programın ara değişken değerleridir. k değişkeni ise programın çıktı değeridir.

Program otomatik türev aracı açısından aşağıdaki şekilde görülecektir ve otomatik türev aracının  $dt_5/dt_1, dt_5/dt_2$  değerlerini hesaplayan bir kod oluşturması beklenmektedir.

$$\mathbf{v} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ k \\ c \end{bmatrix}, \quad \mathbf{C}(\mathbf{v}) = \begin{bmatrix} t_1 - T_1() \\ t_2 - T_2(t_1) \\ \vdots \\ t_n - T_n(t_1, \dots, t_{n-1}) \end{bmatrix} = \begin{bmatrix} t_1 - x \\ t_2 - y \\ t_3 - 2t_2 - 5t_1 - 7 \\ t_4 - t_3 - t_2 - t_1 \\ t_5 - t_4 - t_3 - t_2 - 3t_1 \end{bmatrix} \quad (18)$$

İleri mod için türev hesabı takip eden şekilde olacaktır.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 5 & -2 & 1 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 \\ -3 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \frac{dt_2}{dt_1} \\ \frac{dt_3}{dt_1} \\ \frac{dt_4}{dt_1} \\ \frac{dt_5}{dt_1} \\ -\frac{dt_1}{dt_2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \frac{dt_3}{dt_2} & \frac{dt_3}{dt_1} & 1 & 0 \\ \frac{dt_4}{dt_2} & \frac{dt_4}{dt_1} & \frac{dt_4}{dt_3} & 1 \\ \frac{dt_5}{dt_2} & \frac{dt_5}{dt_1} & \frac{dt_5}{dt_3} & \frac{dt_5}{dt_4} \\ \frac{dt_2}{dt_3} & \frac{dt_2}{dt_4} & \frac{dt_2}{dt_5} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

Bu durumda  $dt_5/dt_1, dt_5/dt_2$  değerlerinin hesaplanabilmesi için OT aracı takip eden doğrusal sistemi çözecektir.

$$\begin{bmatrix} 1 & 0 & 5 & -1 & -3 \\ 0 & 1 & -2 & -1 & -1 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \frac{dt_2}{dt_1} \\ \frac{dt_3}{dt_1} \\ \frac{dt_4}{dt_1} \\ \frac{dt_5}{dt_1} \\ \frac{dt_1}{dt_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (20)$$

Buna karşın ters modda istenilen türev değerlerinin bulunması için aşağıda verilen daha küçük bir doğrusal denklem sistemin çözülmesi yeterli olacaktır.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 5 & -2 & 1 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 \\ -3 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \frac{dt_5}{dt_1} \\ \frac{dt_4}{dt_1} \\ \frac{dt_3}{dt_1} \\ \frac{dt_2}{dt_1} \\ \frac{dt_1}{dt_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (21)$$

Denklem (20) ve (21)'den açıkça görüldüğü üzere, çıktı değerlerinin girdi değerlerine göre hassasiyetini hesaplayabilmek için ileri modda girdi sayısı kadar, ters modda ise çıktı sayısı kadar hesap yeterli olmaktadır. Otomatik türev aracı Tapenade tarafından üretilen türev hesaplayan programda Tablo 3'de verilmiştir.

Tablo 3 Tapenade ile üretilen türev hesaplayan rutinler

İleri Mod	Ters Mod
SUBROUTINE COST_D(x, xd, y, yd, c, cd) IMPLICIT NONE REAL, INTENT(IN) :: x, y REAL, INTENT(IN) :: xd, yd REAL, INTENT(OUT) :: c REAL, INTENT(OUT) :: cd REAL :: z, k REAL :: zd, kd  zd = 2*yd + 5*xd z = 2*y + 5*x + 7 kd = zd + yd + xd k = z + y + x cd = 3*xd + yd + zd + kd c = 3*x + y + z + k  END SUBROUTINE COST_D	SUBROUTINE COST_B(x, xb, y, yb, c, cb) IMPLICIT NONE REAL, INTENT(IN) :: x, y REAL :: xb, yb REAL :: c REAL :: cb REAL :: z, k REAL :: zb, kb  kb = cb zb = kb + cb xb = kb + 5*zb + 3*cb yb = kb + 2*zb + cb cb = 0.0  END SUBROUTINE COST_B

## UYGULAMALAR

Yöntem iki boyutlu, hücre-merkezli, açık, düzensiz çözüm ağı ile sonlu hacimler yöntemi kullanan bir akış çözücü için ayrık adjoint çözücü geliştirilerek denenmiştir. Adjoint yöntemde,  $\frac{\partial R}{\partial x}, \frac{\partial R}{\partial W}, \frac{\partial I}{\partial x}, \frac{\partial I}{\partial W}$  kısmi türev değerlerinin hesaplanması gerekmektedir. Mevcut uygulamada ilgili kısmi türev hesapları otomatik türev yaklaşımı ile gerçekleştirilmiştir. Böylece uygulamada otomatik türev ve ayrık adjoint yöntemin kullanıldığı melez bir yaklaşım benimsenmiştir.

### $\partial R/\partial W$ Matrisinin Hesaplanması

Kısıt fonksiyon değeri sıfıra eşit olmalıdır. Çözücülerde ise değer tam sıfır değerine ulaşamayacaktır. Bu sebeple bu değer artık olarak adlandırılır. Orijinal programda, artık değer hesabı bir seri döngü sonrası olmaktadır. Tüm hücrelerin artık değer hesabı, ancak seri döngüler tamamlandıktan sonra elde edilmektedir. Bu algoritma için ilgili tüm kısmi türev değerlerini otomatik türev yoluyla elde etmek, hesaplama gücü ihtiyacını çok fazla arttıracaktır. Buna karşın,  $\partial R/\partial W$  matrisi doğası gereği seyrek bir yapıdadır. Her hücre için artık değer nispeten az sayıdaki hücre ve hücrenin komşularına bağlıdır. Bu durum esas alınarak, artık değer hesaplayan rutin her bir hücre için bir adımda artık değeri hesaplayacak şekilde değiştirilmiştir. Değiştirilen rutin, hücrenin ve komşu hücrelerin akış değişken değerlerini girdi olarak almakta ve hücre artık değerini hesaplamaktadır. Uygulamada dörtgen hücreler kullanıldığı için rutin, yirmi girdi değişken kabul etmekte ve dört çıktı değeri (artık değerleri) vermektedir.  $\partial R/\partial W$  matrisini hesaplayabilmek adına, ilgili rutin Tapenade ile ileri modda türevi alınmıştır.

Türevi alınan kod, tüm hücreleri kapsayan bir döngünü içine yerleştirilmiştir. Böylece her bir hücrenin artık değerlerinin, akış değişken değerlerine göre türevi hızlıca, sadece hücre ve komşu hücreler gözetilerek hesaplanmaktadır. Hesaplanan matris takip eden şekilde oluşacaktır.

$$\frac{\partial R}{\partial W} = \begin{bmatrix} \frac{\partial R_{1,1}}{\partial W_{1,1}} & \frac{\partial R_{1,1}}{\partial W_{2,1}} & \frac{\partial R_{1,1}}{\partial W_{3,1}} & \frac{\partial R_{1,1}}{\partial W_{4,1}} & \frac{\partial R_{1,1}}{\partial W_{1,2}} & \frac{\partial R_{1,1}}{\partial W_{2,2}} & \dots & \frac{\partial R_{1,1}}{\partial W_{3,n}} & \frac{\partial R_{1,1}}{\partial W_{4,n}} \\ \frac{\partial R_{2,1}}{\partial W_{1,1}} & \frac{\partial R_{2,1}}{\partial W_{2,1}} & \frac{\partial R_{2,1}}{\partial W_{3,1}} & \frac{\partial R_{2,1}}{\partial W_{4,1}} & & & & & \frac{\partial R_{2,1}}{\partial W_{4,n}} \\ \frac{\partial R_{3,1}}{\partial W_{1,1}} & \frac{\partial R_{3,1}}{\partial W_{2,1}} & \frac{\partial R_{3,1}}{\partial W_{3,1}} & \frac{\partial R_{3,1}}{\partial W_{4,1}} & & & & & \frac{\partial R_{3,1}}{\partial W_{4,n}} \\ \frac{\partial R_{4,1}}{\partial W_{1,1}} & \frac{\partial R_{4,1}}{\partial W_{2,1}} & \frac{\partial R_{4,1}}{\partial W_{3,1}} & \frac{\partial R_{4,1}}{\partial W_{4,1}} & & & & & \frac{\partial R_{4,1}}{\partial W_{4,n}} \\ \frac{\partial R_{1,2}}{\partial W_{1,1}} & & & & \ddots & & & & \frac{\partial R_{1,2}}{\partial W_{4,n}} \\ \frac{\partial R_{2,2}}{\partial W_{1,1}} & & & & & \ddots & & & \frac{\partial R_{2,2}}{\partial W_{4,n}} \\ \vdots & & & & & & \ddots & & \vdots \\ \frac{\partial R_{3,n}}{\partial W_{1,1}} & & & & & & & \ddots & \frac{\partial R_{3,n}}{\partial W_{4,n}} \\ \frac{\partial R_{4,n}}{\partial W_{1,1}} & \frac{\partial R_{4,n}}{\partial W_{2,1}} & \frac{\partial R_{4,n}}{\partial W_{3,1}} & \frac{\partial R_{4,n}}{\partial W_{4,1}} & \frac{\partial R_{4,n}}{\partial W_{1,2}} & \frac{\partial R_{4,n}}{\partial W_{2,2}} & \dots & \frac{\partial R_{4,n}}{\partial W_{3,n}} & \frac{\partial R_{4,n}}{\partial W_{4,n}} \end{bmatrix} \quad (22)$$

Yukarıda verilen matriste, ilk indeks ilgili hücrenin artık değer/akış değişken değer vektörünün eleman numarasını, ikinci indeks ise hücre numarasını temsil etmektedir. Elde edilen matris ise  $4N_{hücre} * 4N_{hücre}$  boyutundadır.

### $\partial R/\partial X$ Matrisinin Hesaplanması

Adjoint değişkenleri elde ettikten sonra, amaç fonksiyonunun, çözüm ağı koordinatlarına göre hassasiyet değerlerinin elde edilebilmesi için  $\partial R/\partial X$  kısmi türev değerlerinin hesaplanması gerekmektedir. Bu uygulamada akım hesabı için birinci dereceden bir şema kullanılmıştır. Bu sebeple bir hücrenin artık değer vektörü yalnızca, o hücreyi oluşturan düğüm noktalarının değişiminden etkilenecektir. Düğüm noktasının değişimi sonrası, yüzey normal vektörünün ve yüzey uzunluğunun değişiminden etkilenecektir (23).

$$\sum_{j=1}^{n_{yüzey}} F_{ij} \cdot S_{ij} = R_i \quad (23)$$

Gerekli türev değerlerini hesaplayabilmek için, hücre artık değerlerini hesaplayan rutin, hücreyi oluşturan düğüm noktalarını da girdi olarak alacak ve hücre yüzey normali ve uzunluğunu tekrar hesaplayacak şekilde yeniden yazılmıştır. Ardından bu rutin Tapenade ile türevi alınarak, her bir hücrenin artık değerinin ilgili düğüm noktalarına göre kısmi türevini hesaplayan rutin elde edilmiş ve  $\partial R/\partial X$  matrisi elde edilmiştir. Elde edilen matrisin şekli takip eden şekildedir.

$$\frac{\partial R}{\partial X} = \begin{bmatrix} \frac{\partial R_{1,1}}{\partial X_{1,1}} & \frac{\partial R_{1,1}}{\partial X_{2,1}} & \frac{\partial R_{1,1}}{\partial X_{1,2}} & \frac{\partial R_{1,1}}{\partial X_{2,2}} & \frac{\partial R_{1,1}}{\partial X_{1,3}} & \frac{\partial R_{1,1}}{\partial X_{2,3}} & \dots & \frac{\partial R_{1,1}}{\partial X_{1,m}} & \frac{\partial R_{1,1}}{\partial X_{2,m}} \\ \frac{\partial R_{2,1}}{\partial X_{1,1}} & \frac{\partial R_{2,1}}{\partial X_{2,1}} & \frac{\partial R_{2,1}}{\partial X_{1,2}} & \frac{\partial R_{2,1}}{\partial X_{2,2}} & & & & & \frac{\partial R_{2,1}}{\partial X_{2,m}} \\ \frac{\partial R_{3,1}}{\partial X_{1,1}} & \frac{\partial R_{3,1}}{\partial X_{2,1}} & \frac{\partial R_{3,1}}{\partial X_{1,2}} & \frac{\partial R_{3,1}}{\partial X_{2,2}} & & & & & \frac{\partial R_{3,1}}{\partial X_{2,m}} \\ \frac{\partial R_{4,1}}{\partial X_{1,1}} & \frac{\partial R_{4,1}}{\partial X_{2,1}} & \frac{\partial R_{4,1}}{\partial X_{1,2}} & \frac{\partial R_{4,1}}{\partial X_{2,2}} & & & & & \frac{\partial R_{4,1}}{\partial X_{2,m}} \\ \frac{\partial R_{1,2}}{\partial X_{1,1}} & & & & & & & & \frac{\partial R_{1,2}}{\partial X_{2,m}} \\ \frac{\partial R_{2,2}}{\partial X_{1,1}} & & & & & & & & \frac{\partial R_{2,2}}{\partial X_{2,m}} \\ \vdots & & & & & & & & \vdots \\ \frac{\partial R_{3,n}}{\partial X_{1,1}} & & & & & & & & \frac{\partial R_{3,n}}{\partial X_{2,m}} \\ \frac{\partial R_{4,n}}{\partial X_{1,1}} & \frac{\partial R_{4,n}}{\partial X_{2,1}} & \frac{\partial R_{4,n}}{\partial X_{1,2}} & \frac{\partial R_{4,n}}{\partial X_{2,2}} & \frac{\partial R_{4,n}}{\partial X_{1,3}} & \frac{\partial R_{4,n}}{\partial X_{2,3}} & \dots & \frac{\partial R_{4,n}}{\partial X_{1,m}} & \frac{\partial R_{4,n}}{\partial X_{2,m}} \end{bmatrix} \quad (24)$$

Yukarıda verilen matriste,  $R$  değerlerinin ilk indeks değeri, akış değişkeni değer vektörünün eleman numarasını, ikinci indeks ise hücre numarasını temsil etmektedir.  $X$  değerlerinin, ilk indeks değeri, bir için  $x$ , iki için  $y$  koordinat değerini, ikinci indeks ise düğüm numarasını temsil etmektedir. Oluşan matris ise  $4N_{\text{hücre}} * 2N_{\text{düğüm}}$  boyutundadır.

#### ***$\partial I/\partial W$ ve $\partial I/\partial X$ Matrislerinin Hesaplanması***

Mevcut uygulamada, amaç fonksiyonu,  $I$ , olarak sadece  $CD$  kullanılmıştır. Bu sebepten maliyet fonksiyonunun kısmi değerleri, matris yerine birer vektör oluşturmaktadır. Bunun yanı sıra, maliyet fonksiyonunun kısmi türev değerleri sadece duvara bitişik hücreler için sıfırdan farklı bir değerdedir. İlgili değerlerin hesabı için yakınsamış akış değişken değerleri ve çözüm ağı düğüm koordinatlarını girdi alan "force" rutininin Tapenade ile ters modda türevi alınmış ve değerler hesaplanmıştır. Kısmi türev değerlerini veren vektörlerin görünüşü şu şekildedir.

$$\frac{\partial I}{\partial W} = \left[ \frac{\partial I}{\partial W_{1,1}} \quad \frac{\partial I}{\partial W_{2,1}} \quad \frac{\partial I}{\partial W_{3,1}} \quad \frac{\partial I}{\partial W_{4,1}} \quad \frac{\partial I}{\partial W_{1,2}} \quad \frac{\partial I}{\partial W_{2,2}} \quad \dots \quad \frac{\partial I}{\partial W_{3,n}} \quad \frac{\partial I}{\partial W_{4,n}} \right] \quad (25)$$

$$\frac{\partial I}{\partial X} = \left[ \frac{\partial I}{\partial X_{1,1}} \quad \frac{\partial I}{\partial X_{2,1}} \quad \frac{\partial I}{\partial X_{1,2}} \quad \frac{\partial I}{\partial X_{2,2}} \quad \frac{\partial I}{\partial X_{1,3}} \quad \frac{\partial I}{\partial X_{2,3}} \quad \dots \quad \frac{\partial I}{\partial X_{1,m}} \quad \frac{\partial I}{\partial X_{2,m}} \right] \quad (26)$$

$\partial I/\partial W$  vektörü  $1 * 4N_{\text{hücre}}$ ,  $\partial I/\partial X$  vektörü ise  $1 * 2N_{\text{düğüm}}$  boyutundadır.

#### ***Adjoint Çözücü***

Önceden belirtildiği üzere, adjoint denklem takip eden şekildedir.

$$\underbrace{\left[ \frac{\partial R}{\partial W} \right]^T}_{4N_{\text{hücre}} * 4N_{\text{hücre}}} \lambda = - \underbrace{\left[ \frac{\partial I}{\partial W} \right]^T}_{4N_{\text{hücre}} * 1} \quad (27)$$

Denklem (27) ile elde edilen  $\lambda$  adjoint vektörü  $4N_{\text{hücre}} * 1$  boyutunda olacaktır. Önceden de değinildiği üzere artık değerlerin jacobian matrisi, doğası gereği seyrektiler. Bu sebepten, sistemi en verimli şekilde çözebilmek adına, Intel Math Kernel Library PARDISO (paralel direk çözücü)



rutinlerinden faydalanılmıştır. Matrisler bellekte az yer tutmaları için sıkıştırılmış seyrek satır biçiminde saklanmıştır.

### Toplam Hassasiyet Denklemi

Adjoint vektör Intel MKL PARDISO rutinleri ile çözüldükten sonra, toplam hassasiyet denklemi çözümlenerek sürükleme kuvveti katsayısının, çözüm ağı koordinatlarına göre hassasiyeti hesaplanmıştır.

$$\underbrace{\frac{dCD}{dx}}_{1 * 2N_{dügüm}} = \underbrace{\frac{\partial CD}{\partial x}}_{1 * 2N_{dügüm}} + \underbrace{\lambda^T \frac{\partial R}{\partial x}}_{4N_{hücre} * 2N_{dügüm}} \quad (28)$$

↓  
1 \* 4N<sub>hücre</sub>

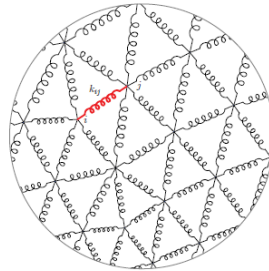
Denklem (28) adjoint vektör ile artık değerlerin jacobian matrisinin çarpımının hesaplanmasını gerektirmektedir. Burada da verimli bir hesaplama için, vektör matris çarpımında Intel MKL gömülü çözümlerinden SCSRGEMV rutini kullanılmıştır. Ardından bu çarpım sonucu elde edilen vektör ile sürükleme katsayısının çözüm ağı koordinatlarına kısmi hassasiyetini veren vektör ile toplanarak, sürükleme katsayısının çözüm ağı koordinatlarına göre toplam hassasiyet değerleri elde edilmiştir.

### Çözüm Ağı Deformasyon Yöntemi

Geometriyi tanımlayan kontrol parametrelerinin belirlenen amaç fonksiyonuna göre hassasiyet değerlerini elde edebilmek için " $dI/d\alpha$ ", geometri değişimi ile tüm çözüm ağı düğüm noktalarının değişiminin bilinmesi " $dX/d\alpha$ " ve adjoint çözümleri ile elde edilen çözüm ağı düğüm noktalarının amaç fonksiyonuna hassasiyeti " $dI/dX$ " ile çarpılması gerekmektedir. Bu yüzden geometri değişimi ile çözüm ağını deforme eden bir rutin geliştirilmiş ve geliştirilen rutin ileri moda türevi alınarak birim geometri değişiminin her bir çözüm ağı düğüm noktasına etkisi hesaplanmıştır.

$$\frac{dI}{d\alpha} = \frac{dI}{dX} \cdot \frac{dX}{d\alpha} \quad (29)$$

Mevcut çalışmada çözüm ağı deformasyonu için yay örnekesmesi [Batina, 1991] yöntemi kullanılmıştır. "Bu teknikte birbirine bağlı çözüm ağı noktaları, aralarındaki hayali yaylar, geometrinin değişimine bağlı olarak, çözüm ağı noktaları üzerinde yaylanma kuvveti uygulamakta ve bu kuvvetlerin çözüm ağı noktalarını yeni denge konumuna getirecek şekilde hareket ettirmesi sonucunda yeni çözüm ağı elde edilmektedir" [Akgül, 2008].



Şekil 2 Yay Örnekesmesi şematik gösterimi [Akgül, 2008]

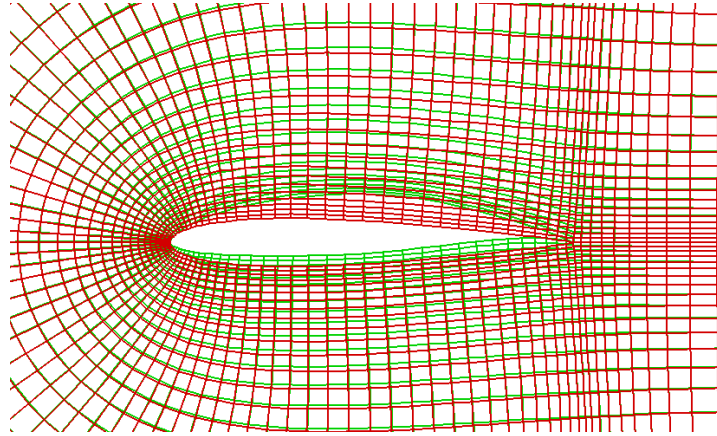
Bu yöntemde, yay sertlik değerleri, ilgili kenar uzunluğunun tersi olarak hesaplanır. Takip eden denklemler, yay sertlik değeri hesabı ve denge durumu hesabı için kullanılmaktadır.

$$k_{ij} = \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \quad (30)$$

$$\Delta x_i^{n+1} = \frac{\sum_m k_m \Delta x_i^n}{\sum_m k_m}, \Delta y_i^{n+1} = \frac{\sum_m k_m \Delta y_i^n}{\sum_m k_m} \quad (31)$$

$$x_i^{n+1} = x_i^n + \Delta x_i^{n+1}, y_i^{n+1} = y_i^n + \Delta y_i^{n+1} \quad (32)$$

Aşağıdaki şekilde (Şekil 3), geliştirilen rutinün kabiliyetini göstermek adına NACA 0012 profili için oluşturulan çözüm ağının, başka bir kanat kesiti için rutin tarafından deforme edilmiş çözüm ağı ile kıyaslaması sunulmuştur.

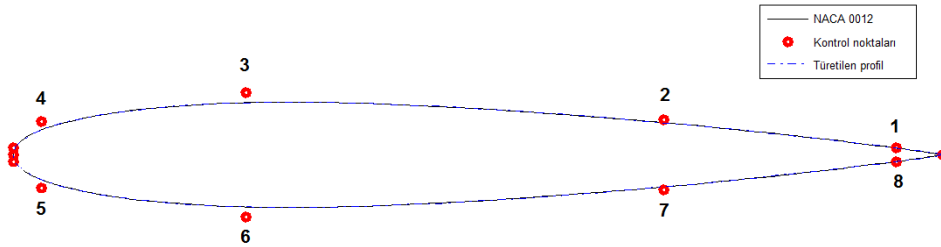


Şekil 3 NACA 0012 Çözüm ağının yazılan rutin ile başka bir profile adapte edilmesi

### SONUÇ

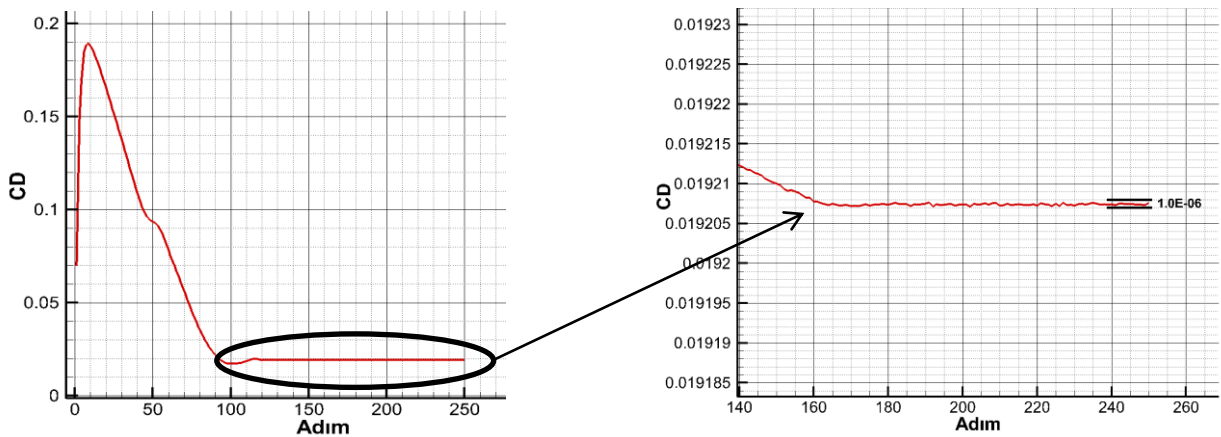
Geliştirilen adjoint çözücünün kabiliyetini göstermek adına, akış çözücü ile NACA 0012 profili üzerinde akış analiz gerçekleştirilmiş ve ardından adjoint çözücü ile sürüklenme katsayısı için hassasiyet değerleri hesaplanmıştır.

Çalışmada profil B-spline eğrileri ile modellenmiş ve profil 12 kontrol noktası ile tanımlanmıştır. Kontrol noktaları, B-spline eğrileri ile türetilen profil ve NACA 0012 profilinin arasındaki farkı eniyileme ile asgariye düşürerek belirlenmiştir.



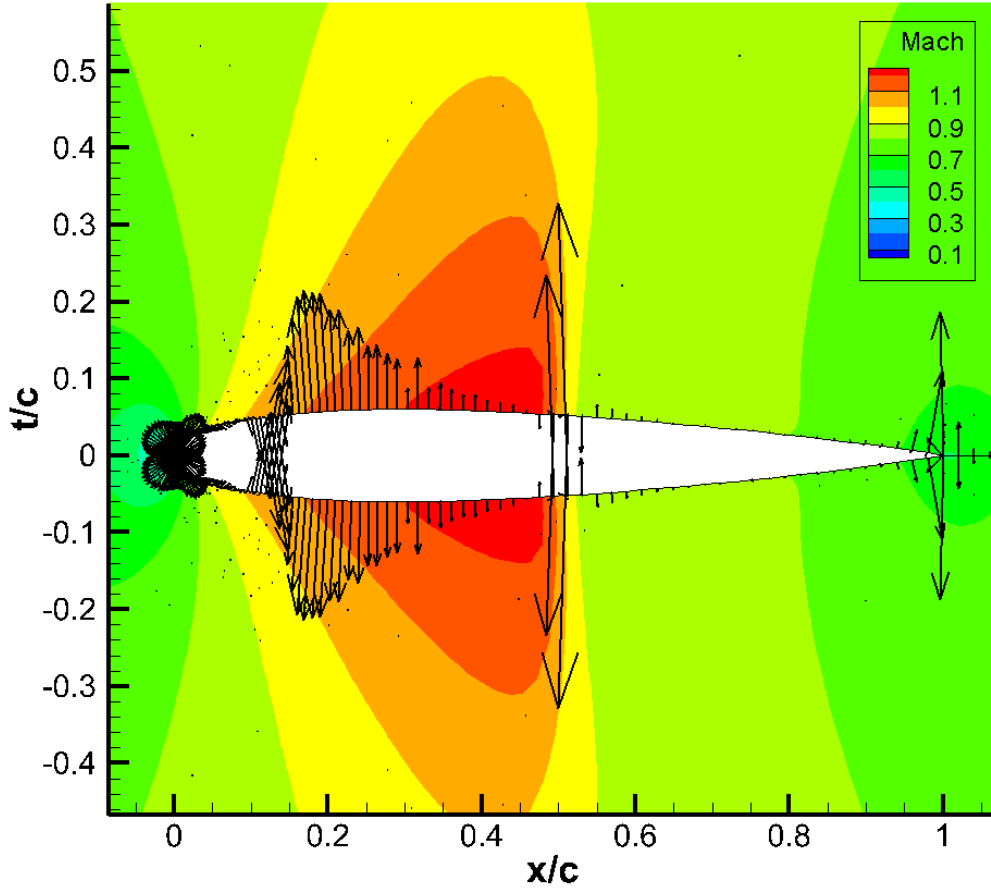
Şekil 4 Kontrol noktaları ve B-spline eğrileri ile modellenen profilin NACA 0012 profili ile kıyaslaması

Test koşulu olarak NACA 0012 profili 0,8 Mach sayısında, 0 derece hücum açısı için analiz edilmiştir. Analizlerde çözüm ağı olarak dörtgen elemanlardan oluşan ve  $480 \times 120$  düğüm noktasına sahip bir çözüm ağı kullanılmıştır. CFL sayısı olarak, her iki adımda 1,1 kat artacak şekilde, başlangıçta 1.0 değeri kullanılmıştır. CFL sayısı 10000 değerini geçince limitlenmiştir. Toplamda 250 adım analiz yapılmıştır. Sürüklenme katsayısı yakınsama grafiği Şekil 5'de verilmiştir.



Şekil 5 Sürüklenme kuvveti katsayısı yakınsama grafiği

Çözümde yakınsama elde edildikten sonra, adjoint çözücü ile sürüklenme kuvveti katsayısı hassasiyet değerleri hesaplanmıştır (Şekil 6) . Şekil 6 ile verilen vektörler her bir çözüm ağı düğüm noktasının sürüklenme kuvveti katsayısını ne kadar değiştireceğini temsil etmektedir.



Şekil 6 Ses hızı renk haritası ve çözüm ağı üzerinde düğüm noktalarının CD hassasiyetleri,  $dCD/dX$

Adjoint çözücü ile elde edilen hassasiyet değerlerinin teyidi için firar kenarı ve hücum kenarını temsil eden kontrol noktaları hariç, diğer kontrol noktalarının sürüklenme kuvveti katsayısına etkisi değerlendirilmiştir. Bu kapsamda geometriyi tanımlayan kontrol noktalarının veter boyunca konumunu değiştirmeden, y ekseninde birim hareketinin sürüklenme kuvveti katsayısına etkisi, hem adjoint yöntem ile hem de kontrol noktalarına 0.001 birim veter değerinde y eksenini yönünde değişim verilerek sonlu farklar yöntemi ile hesaplanmıştır. Değerlerin kıyaslaması Tablo 4'de sunulmuştur.

Tablo 5 Adjoint yöntem ile sonlu farklar yöntemi sonuçları

Kontrol Noktası	Adjoint	Sonlu Farklar ( $\Delta = 0.001$ )	Fark (%)
1	0.061	0.062	-1.61
2	-0.164	-0.166	-1.20
3	0.553	0.554	-0.18
4	0.160	0.161	-0.62
5	-0.151	-0.148	2.03
6	-0.542	-0.535	1.31
7	0.162	0.170	-4.71
8	-0.022	-0.021	4.76

Sonuçlar değerlendirildiğinde sonlu farklar yöntemi kullanılarak elde edilen gradyan değerleri ile adjoint yöntem kullanılarak elde edilen gradyan değerleri arasında görülen azami farkın % 5'in altında olmuştur. Adjoint yöntemin kesin türev değerlerini verdiği, sonlu farkların ise ancak delta

değerin sıfır olduğu koşulda kesin türev değerlerini vereceği göz önüne alınırsa, belirlenen farklar kabul edilebilir seviyede olarak değerlendirilmiştir.

### DEĞERLENDİRME

Çalışma neticesinde otomatik türev araçları kullanılarak, nispeten az efor ile adjoint çözücü geliştirilebileceği gösterilmiştir. Çalışmada uygulanan yöntem, kolaylıkla ağdalı akışlar içinde uygulanabilecektir. Çalışmanın devamında ağdalı akışlar için adjoint çözücü geliştirilmesi ve adjoint çözücü ile hesaplanan gradyanlarla eniyileme çalışması yapılması hedeflenmektedir. Bir sonraki aşama olarak da, belirlenen amaç fonksiyonu için akış çözücü ile elde edilen değerleri, adjoint çözücünden elde edilecek gradyan bilgileri ile destekleyerek, az sayıda çözüm ile elde edilebilecek yüksek doğrulukta bir vekil model [Bouhlel, 2018] üzerinden evrensel eniyileme yapılması amaçlanmaktadır.

### Kaynaklar

- [1] Akgül, A., Tiftikçi H., 2008. “Hesaplamalı Akışkanlar Dinamiği ile Aerodinamik Tasarım En iyilemesi”, II. Ulusal Havacılık ve Uzay Konferansı, UHUK-2008-079
- [2] Batina, J.T., 1991. “Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex Aircraft Aerodynamic Analysis”, AIAA Journal, Vol. 29, No.3, pp. 327-333.
- [3] Bouhlel, Mohamed A. ve Martins, Joaquim R. R. A., 2018. “Gradient-Enhanced Kriging for High-Dimensional Problems”, Engineering with Computers, <https://doi.org/10.1007/s00366-018-0590-x>
- [4] Giles, M. B., Duta, M. C., Müller, J. D. ve Pierce N. A., 2003. “Algorithm Developments for Discrete Adjoint Methods”. AIAA Journal, 41(2):198–205.
- [5] Griewank, A., 2000. “Evaluating Derivatives”. SIAM, Philadelphia.
- [6] Hascoet, L. ve Pascual, V., 2004. “Tapenade 2.1 User's Guide”. Technical report 300, INRIA. URL, <http://www.inria.fr/rrrt/rt-0300.html>.
- [7] Martins, Joaquim R. R. A. ve Hwang, J. T., 2013. “Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models”. AIAA Journal, 51(11):2582-2599.