

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**TAŞIYICI YÜZEY SİSTEMLERİNİN
GİRDAP KAFES YÖNTEMİ İLE ANALİZİ
VE GENETİK ALGORİTMA İLE ENİYİLEMESİ**

YÜKSEK LİSANS TEZİ

Osman Mirza DEMİRCAN

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

HAZİRAN 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**TAŞIYICI YÜZEY SİSTEMLERİNİN
GİRDAP KAFES YÖNTEMİ İLE ANALİZİ
VE GENETİK ALGORİTMA İLE ENİYİLEMESİ**

YÜKSEK LİSANS TEZİ

**Osman Mirza DEMİRCAN
511161140**

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Prof. Dr. M. Adil YÜKSELEN

HAZİRAN 2018

İTÜ, Fen Bilimleri Enstitüsü'nün 511161140 numaralı Yüksek Lisans Öğrencisi Osman Mirza DEMİRCAN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "TAŞIYICI YÜZEY SİSTEMLERİNİN GİRDAP KAFES YÖNTEMİ İLE ANALİZİ VE GENETİK ALGORİTMA İLE ENİYİLEMESİ" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. M. Adil YÜKSELEN**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Yrd. Doç. Dr. Nazım Kemal ÜRE**
İstanbul Teknik Üniversitesi

Prof. Dr. O. Ergüven VATANDAŞ
İstanbul Gelişim Üniversitesi

Teslim Tarihi : 7 Mayıs 2018
Savunma Tarihi : 5 Haziran 2018





Canım aileme,



ÖNSÖZ

Akademik eğitimim boyunca farklı türde hava araçlarının tasarımı ve analizi üzerine çeşitli dersler aldım. Ders dışı katıldığım ulusal ve uluslararası öğrenci yarışmalarında bu eğitimimi uygulamaya dökme şansı buldum. Bu süreçte tasarıma olan merakım gelişti ve tasarım üzerine bilgisayar programları yazmaya başladım.

Tasarımlarımda, aldığım eğitimin yanı sıra, doğadan da örneklerle bakarak çıkarımlar yaptığım oldu. Doğadaki eniyileme mekanizması olan evrimin bilgisayar ortamında uygulaması sayesinde, hava araçlarının tasarım ve eniyileme sürecinin daha kolay ve hızlı bir şekilde gerçekleştirilebildiğini öğrendim. Yazdığım bu tez kapsamında da hava araçlarının hesaplamalı aerodinamik yöntemleriyle analizi ve genetik algoritmaların kullanımı ile eniyilemesini konu aldım.

Hayatımın her anında yardımına koşup beni motive eden aileme, karşılaştığım zorluklar karşısında beni destekleyen arkadaşlarıma ve hocalarıma minnettarım. Özellikle, bu çalışmayı hazırlarken bana desteklerini esirgemeyen, engin bilgi ve tecrübeleriyle bana yol gösteren danışmanım Sayın Prof. Dr. M. Adil Yükselen'e teşekkürü borç bilirim.

Mayıs 2018

Osman Mirza Demircan
Uçak Mühendisi



İÇİNDEKİLER

Sayfa

ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR.....	xi
SEMBOLLER.....	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET	xix
SUMMARY	xxi
1. GİRİŞ	1
1.1 Motivasyon.....	1
1.2 Literatür Araştırması.....	2
1.2.1 Girdap Kafes Yöntemi	2
1.2.2 Genetik Algoritma	5
2. GİRDAP KAFES YÖNTEMİ	9
2.1 Çok Elemanlı Taşıyıcı Yüzey Sistemleri İçin Girdap Kafes Yöntemi.....	9
2.2 Aerodinamik Katsayıların Hesaplanması	13
2.3 Denklem Takımındaki Katsayıların Hesaplanması	16
2.4 Programlama	17
2.4.1 Ön işleme.....	18
2.4.2 İşleme	21
2.4.3 Son işleme	22
2.5 Doğrulama Çalışmaları.....	22
2.5.1 PLL ile karşılaştırma	22
2.5.2 Yer etkisi analizi	24
2.5.3 Kanat-kuyruk sistemi analizi.....	26
3. GENETİK ALGORİTMA.....	29
3.1 Taşıyıcı Yüzey Sistemi Eniyilemesi İçin Darvinci Evrim Teorisi	29
3.2 Programlama	31
3.2.1 Rastgele birey oluşturma	32
3.2.2 Bireylerin değerlendirilmesi.....	33
3.2.3 Durma kriterleri	34
3.2.4 Çaprazlama ve mutasyon.....	34
3.2.5 Gen aktarımı.....	35
4. UYGULAMALAR	39
4.1 Kanat Eniyilemesi	39
4.2 Kanat-Kuyruk Sistemi Eniyilemesi	42
5. SONUÇ VE ÖNERİLER.....	47
KAYNAKLAR	51
ÖZGEÇMİŞ	69



KISALTMALAR

CFD	: Hesaplamalı Akışkanlar Dinamiği (Computational Fluid Dynamics)
DET	: Darvinci Evrim Teorisi (Darwinian Evolution Theory)
EA	: Evrimsel Algoritma
GA	: Genetik Algoritma
NACA	: Amerikan Ulusal Havacılık Danışma Kurulu (National Advisory Committee for Aeronautics)
NASA	: Amerikan Ulusal Havacılık ve Uzay Dairesi (National Aeronautics and Space Administration)
NLL	: Nümerik Taşıyıcı Çizgi (Numerical Lifting Line)
PLL	: Prandtl Taşıyıcı Çizgi (Prandtl Lifting Line)
VLM	: Girdap Kafes Yöntemi (Vortex Lattice Method)



SEMBOLLER

A_{ij}	: Katsayılar matrisi
D_i	: İndüklenmiş sürüklenme kuvveti
L	: Taşıma kuvveti
M_x, M_y, M_z	: Moment bileşenleri
R_i	: Sağ taraf matrisi
S	: Planform alan
V	: Hız vektörü
AR	: Açıklık oranı
NI	: Veter boyunca panel sayısı
NJ	: Açıklık boyunca panel sayısı
NK	: Taşıyıcı yüzey sayısı
g	: Nesil
h	: İrtifa
n	: Panel normali
q	: Dinamik basınç
v	: İndüklenmiş hız vektörü
Γ	: Girdap şiddeti
α	: Hücüm açısı
λ	: Sivrilme oranı
ρ	: Yoğunluk



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Tek-kanat için doğrulama testinde PLL uygulaması için girdiler.	23
Çizelge 2.2 : Tek-kanat doğrulama testi için PLL ve VLM uygulamalarından elde edilen aerodinamik katsayılar.	23
Çizelge 2.3 : Kanat-kuyruk doğrulaması için oluşturulan kanadın özellikleri.	26
Çizelge 2.4 : Kanat-kuyruk doğrulaması için oluşturulan kuyruğun özellikleri.	27
Çizelge 2.5 : Kanat-kuyruk sisteminin analizinden elde edilen sonuçlar.	27
Çizelge 3.1 : Örnek bir popülasyon için uygunluk değeri çizelgesi.	30
Çizelge 3.2 : Örnek değişken gen parametreleri.	33
Çizelge 3.3 : Örnek bir popülasyona ait normalize uygunluk ve baskınlık değerleri.	35
Çizelge 3.4 : Farklı gen aktarma yöntemlerinin karşılaştırması.	36
Çizelge 4.1 : Kanat eniyilemesi için verilen orijinal parametreler.	39
Çizelge 4.2 : Kanat eniyilemesi için değişim aralıkları.	41
Çizelge 4.3 : Orijinal ve en iyi kanat karşılaştırması.	42
Çizelge 4.4 : Kanat-kuyruk eniyilemesinde kuyruk için değişim aralıkları.	43
Çizelge 4.5 : Orijinal ve eniyilenmiş kanat-kuyruk sistemi karşılaştırması.	45



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Prandtl taşıyıcı çizgi modeli (Katz ve Plotkin, 2001).....	3
Şekil 1.2 : Nümerik taşıyıcı çizgi modeli.....	4
Şekil 1.3 : Çok elemanlı kanat yüzeyi için girdap-kafes modeli (DeYoung, 1976). ...	5
Şekil 1.4 : Genetik algoritmalar için genel akış çizelgesi.	7
Şekil 2.1 : Çok elemanlı taşıyıcı yüzey sistemi.....	9
Şekil 2.2 : Taşıyıcı yüzeyin açıklık ve veter doğrultusunda panellere ayrılması ve atnalı girdaplarının yerleşimi.	10
Şekil 2.3 : Çeyrek-üççeyrek kuralına göre panel üzerinde atnalı girdabı ve kontrol noktası yerleşimi.	10
Şekil 2.4 : Bir atnalı girdabının bir kontrol noktasındaki etkisi.	11
Şekil 2.5 : Herhangi bir panel üzerindeki etkin hız ve aerodinamik kuvvet vektörleri.	14
Şekil 2.6 : Herhangi bir panel üzerinde oluşan aşağı sapma.	14
Şekil 2.7 : Taşıma ve indüklenmiş sürüklenme doğrultularındaki birim vektörler.	15
Şekil 2.8 : Panel normaline için köşegen çarpımı gösterimi.....	16
Şekil 2.9 : Kontrol noktasında indüklenen hız vektörü (Biot-Savart Kanunu).	16
Şekil 2.10 : VLM programı genel akış çizeneği.	17
Şekil 2.11 : Uzaydaki konumuna yerleştirilen panellere bölünmüş taşıyıcı yüzey... 20	
Şekil 2.12 : Kanadın paneller halinde gösterimi (birimler metre cinsindedir).	23
Şekil 2.13 : Doğrulama testi için PLL ve VLM sonuçlarının karşılaştırması.	24
Şekil 2.14 : Yer etkisi analizi arayüzü.	25
Şekil 2.15 : Yer etkisi analizi sonuçları.	25
Şekil 2.16 : Kanat-kuyruk doğrulaması için oluşturulan kanat üzerindeki panel ve atnalı girdapları.	26
Şekil 2.17 : Kanat-kuyruk doğrulaması için oluşturulan taşıyıcı yüzey sistemi üzerindeki paneller.	27
Şekil 3.1 : Yapay bir sistem için genotip ve fenotip gösterimi.	29
Şekil 3.2 : Çarkı çevir yöntemi.	30
Şekil 3.3 : Çaprazlama ve mutasyon evreleri gösterimi.....	31
Şekil 3.4 : Taşıyıcı yüzeylerin eniyilemesi için akış çizelgesi.	32
Şekil 4.1 : Kanat eniyilemesi için uygunluk fonksiyonu.	40
Şekil 4.2 : Kanat eniyilemesi uygunluk grafiği.....	41
Şekil 4.3 : Kanat eniyilemesinden elde edilen son geometri (kesikli çizgi orijinal kanattır).	42
Şekil 4.4 : Kanat-kuyruk eniyilemesi uygunluk grafiği.	44
Şekil 4.5 : Kanat-kuyruk eniyilemesinden elde edilen son geometri (kesikli çizgi orijinal kanat-kuyruk sistemidir).	44



TAŞIYICI YÜZEY SİSTEMLERİNİN GİRDAP KAFES YÖNTEMİ İLE ANALİZİ VE GENETİK ALGORİTMA İLE ENİYİLEMESİ

ÖZET

Hava araçlarının aerodinamik tasarımı, analizi ve eniyilemesi, akışkanlar dinamiği bilimiyle uğraşan araştırmacıların üzerinde durduğu karmaşık ve esasında çok-disiplinli mühendislik problemleridir. Bu kapsamda, günümüzde uygulanan Hesaplamalı Akışkanlar Dinamiği (CFD) çözümleri tasarım sürecinin ilk aşamalarında birçok sebeple uygun görülmemektedir. Onun yerine daha basit çözümler sunan, akış alanıyla ilgili temel yaklaşımlarda bulunarak çözümü hızlandıran yöntemlerin kullanımı daha yaygındır.

Hesaplamalı aerodinamik uygulamalarında sıkça kullanılan Girdap Kafes Yöntemi (VLM) taşıyıcı yüzeylerin üzerindeki potansiyel akışı çözmektedir. Ludwig Prandtl'ın 1918 yılında tanıttığı taşıyıcı çizgi modeli (PLL) sonlu düz kanatlar için analitik bir çözüm sağlarken, yine Prandtl'ın ortaya koyduğu teoriyi esas alan VLM ile herhangi bir konfigürasyondaki çok elemanlı taşıyıcı yüzey sisteminin incelemesi girdap-kafes modeli ile mümkün olmaktadır. Bu modelde yüzeyler çözümlemenin yapılacağı panellere bölünür ve her panele taşıyıcı özelliğini veren atnalı girdapları yerleştirilir. Potansiyel akım sınır şartı kullanılarak çözülen girdap şiddetleri ile taşıyıcı yüzey üzerine etki eden aerodinamik kuvvetler Kutta-Joukowski Kanunu ile hesaplanır.

Konsept ve ön tasarım süreçlerinde ilgili hava aracının taşıyıcı yüzeyleriyle ilgili kanat açıklığı, açıklık oranı, kök ve uç veter uzunlukları gibi birçok parametrenin belirlenmesi gerekmektedir. Ancak, her bir parametrenin belirlenmesi başlı başına bir eniyileme çalışması gerektirmektedir. Bunun sebebi ise taşıyıcı yüzeyler birbirleri arasında aerodinamik etkileşim içindedir. Bir yüzeye ait parametre diğer bir yüzey üzerinde etki yaratmaktadır ve hava aracının genel performansı açısından bu etkinin incelenmesi önem arz etmektedir.

Doğanın genetik mekanizmalarını açıklayan Darvinci Evrim Teorisi (DET), yapay sistemlerin eniyilemesinde de etkin bir şekilde kullanılabilir. Evrimsel Algoritmaların (EA) bir alt sınıfı olan Genetik Algoritmalar (GA), doğanın kanunlarını yapay sistemlere uygulayarak, zorluklar karşısında kendi kendini geliştiren sistemler üretmektedir. Bu uygulamanın temellerini John H. Holland 1975 yılında atmıştır. Holland'ın yürüttüğü çalışmalar hem yapay sistemlerin eniyilemesi hem de doğal sistemlerin bilimine büyük katkı sağlamıştır.

Bu tez kapsamında çok elemanlı taşıyıcı yüzeylerin nümerik incelemesi için VLM yönteminin bir bilgisayar programını geliştirilmiştir. MATLAB programlama dili kullanılarak geliştirilen yazılımın öncelikle PLL yöntemi ile düz bir kanat analizi için karşılaştırılmış ve elde edilen sonuçlar doğrulanmıştır. Çok elemanlı sistemler içinse ek doğrulama çalışmaları yürütülmüştür. Ters açılı bir çift kanat kullanılarak yer etkisi analizi yapılmış ve kanatlar arası uzaklık artırıldıkça yer etkisinin azaldığı görülmüştür. Bir diğer doğrulama çalışmasında ise bir kanat arkasında yerleştirilen

yatay kuyruğun analizi yapılmıştır. Bu analizde aşağı sapmanın etkisiyle kuyruk üzerindeki etkin hücum açısının serbest akımın geliş açısından daha düşük hesaplanması incelenmiştir.

Doğrulama çalışmaları neticesinde, taşıyıcı yüzeylerin incelenmesi amacıyla geliştirilen VLM yazılımı evrimsel döngüler sırasında çağırılacak bir fonksiyona dönüştürülmüştür. Evrimsel döngülerin değerlendirme aşamasında, her bir birey için bu fonksiyon çağırılmış ve aerodinamik katsayılar hesaplanmıştır. Hesaplanan katsayılar güçlü bireylerin belirlenmesine yarayan uygunluk fonksiyonlarında kullanılmıştır. Çaprazlama ve mutasyon evrelerinden sonra elde edilen çocuk bireylerin sonraki nesle nasıl aktarılacağı konusunda farklı yaklaşımlar denenmiş olup, elde edilen sonuçlar karşılaştırılmıştır. Sonuç olarak geliştirilen taşıyıcı yüzey sistemlerini eniyileme yazılımı uygulamalarda kullanılmış ve elde edilen bulgular paylaşılmıştır.



ANALYSIS AND OPTIMIZATION OF LIFTING SURFACE SYSTEMS USING VORTEX LATTICE METHOD WITH GENETIC ALGORITHM

SUMMARY

The design and optimization of aircraft has been a center of interest for researchers working in the field of fluid dynamics. The science of aircraft design has always been a multidisciplinary subject and therefore, a rather complex field of research. The analysis process of aircraft is nowadays carried out by means of Computational Fluid Dynamics (CFD), which requires a great amount of effort and resources. In consequence, simpler methods based on some basic assumptions about the fluid domain are employed during the concept or preliminary design phase. Methods that solve the inviscid flow are essentially predominant.

In the past century, extensive research has been made on methods based on the potential flow. Ludwig Prandtl and his colleagues from Göttingen, Germany introduced Prandtl's Lifting Line (PLL) Theory in 1918. Lifting surfaces are modelled as lines, hence the name of the method. PLL proved to be invaluable for the analysis of finite wings. Nevertheless, the theory can only be used with straight wings (no sweep or dihedral angle) having only one lifting surface. Moreover, numerical derivatives of the method are developed in order to overcome its deficiencies.

For instance, the Vortex Lattice Method (VLM) has become increasingly popular in the 1960's and it's still used today for analyzing lifting surfaces. In PLL, a lifting line is an infinite number of nested horseshoe vortices with the bound vortices passing through the quarter chord of the surface and the trailing vortices following the surface until the trailing edge, where they go to infinity parallel to the freestream. The induced velocities by these vortices are calculated at the three-quarters of the surface sections. This quarter-three-quarter method is the fundamental rule of panel methods which arises from the Kutta condition. In VLM, wings are discretized as a lattice of panels with horseshoe vortices on each individual panel, unlike an infinite number of nested horseshoe vortices in PLL theory. The vortex strengths on these panels are computed using the potential flow boundary condition: the velocity component normal to the surface should be zero. After calculating the vortex strengths, the Kutta-Joukowski Theorem is used in order to calculate the acting aerodynamic forces on the lifting surface. Finally, the lift coefficient and the induced drag coefficient can be calculated.

In the concept and preliminary phases of aircraft design, several parameters such as the span, aspect ratio, root and tip chord lengths have to be determined. These parameters cannot be considered separately since determining each parameter affects the other regarding the overall performance of the aircraft. An optimization has to be made for each parameter, which sometimes requires making a compromise. Hence, this process has to be handled carefully. Current optimization methods offer various solutions while having limitations on the search space concerning continuity, existence of derivatives and other matters.

Selecting and applying the right optimization method can be a tedious task. However, one type of optimization is more flexible for varying requirements compared to others: Genetic Algorithms (GA). Genetic algorithms are stochastic searching algorithms known as a subclass of Evolutionary Algorithms (EA).

The laws of nature described in the Darwinian Evolution Theory (DET) drive GAs. Features of self-guidance, development and reproduction are the rules of genetics. These rules when applied to artificial systems have the potential to create the fittest product for a given objective. The pioneering experiments on GA introduced by John H. Holland and his colleagues at the University of Michigan in 1975 not only contributed to the science of artificial systems but also biological systems. GA employs the genetic mechanism of the nature through three main principles: heredity, variation and selection. A very robust tool can be created by studying each of these principles and incorporating them in an optimization framework.

First, a computer program for the analysis of lifting surface systems is developed using the formulations in the VLM method. The MATLAB programming language is used since it offers many intrinsic functions and tools to generate graphic interfaces. The VLM code is then put to test by analyzing a straight wing and the results are compared with the PLL output. Another analysis is carried out using symmetrically placed lifting surfaces. The ground effect is then simulated on the surfaces by changing the distance between them. Lastly, the downwash effect on the horizontal tail behind a wing is analyzed and the decreasing effective angle of attack on the tail surface is observed.

After the validation process, the VLM code is transformed into a function and implemented into another code which carries out evolutionary loops in order to optimize the lifting surface by means of GA.

First of all, a random population is generated and analyzed using the VLM function. The code then uses fitness functions with the aerodynamic coefficients as inputs in order to determine the strongest specimen. If the variation is no longer satisfied, the evolutionary loop is broken. This is handled by checking the mean fitness curve of the population at each generation. If the curve's slope reaches zero, it means that the variation principle is compromised.

A mating pool is created in which the strong specimen having a better chance of reproducing. Parent genes are used in a process called "crossover" and by consequence, an offspring is produced. The offspring must have traits coming from both their parents. Additionally, the offspring are sometimes subjected to mutation which can completely alter their genes. At the end of each generation, the offspring are added to the population while weak specimen are removed. This process forces the population to produce better specimen by removing defective genes and developing better ones.

The way the offspring are added to the next generation is another matter. In this work, three different ways of handling this process are analyzed: inheritance, controlled inheritance and survival. In the first method, the offspring takes the place of its parents. In the controlled inheritance method, the offspring is only added to the population if its fitness is greater than its parents. In the last method, the offspring takes the place of the weakest specimen in the population and if its fitness is even lesser, the offspring is discarded. The first method results in an everchanging mean fitness curve while other converge towards a fitness value.

The GA code is used in two applications in order to fully demonstrate its capabilities. An arbitrary wing design is defined in the code with a goal lift coefficient. A fitness function is written with values closer to the goal lift coefficient returning a higher value. After entering its limitations, the wing is added to a randomly generated wing population with the same goal. After several generations, a better wing is obtained with a greater fitness value.

In the following application, a horizontal tail is placed behind the optimized wing and the longitudinal stability requirement is added to the fitness function. The genes of both lifting surfaces are subjected to evolutionary process and the resulting fittest specimen is analyzed. The addition of the tail to the system and the longitudinal stability requirement with it, forced the previously optimized wing to change and adapt.

In conclusion, the use of GAs in optimization of lifting surface systems offers a robust multi-objective design framework. This thesis only conducts aerodynamic analysis and checks for simple aerodynamics and stability requirements. However, any type of requirement and related analysis procedure can be added to the process without making major changes to the optimization framework. This is essentially why GAs are widely used in the field of design and optimization, they are flexible and compatible with any change or addition to evaluation process of the design parameters.

The use of GAs are mostly encouraged for the optimization procedure of very complex systems where a great amount of variables is present. Considering all these variables are to be handled in a population of specimen, calculating the fitness of each specimen at each generation is significantly time consuming for problems of higher order. Therefore, the computers of today are only capable of optimizing relatively simple problems in the matter of preliminary design of artificial systems. However, the future of computer science can overcome this issue of time and resource consumption since the quantum computers hold a great potential.



1. GİRİŞ

1.1 Motivasyon

Bilgisayar çağı öncesinde havacılıktaki gelişmeler genellikle elde mevcut olan uçakların ölçülen performanslarından hareketle, geometrileri, malzemeleri ve donanımları üzerinde nispeten küçük değişiklikler yaparak sağlanmıştır. Bunun en önemli nedeni bir uçağın tasarımını sırasında üzerinde değişiklik yapılabilecek yüzlerce parametre olması, fakat bu parametreler üzerinde değişiklikler yapıldığında gerekli olan analizlerin kısa sürede gerçekleştirilmesine imkân olmayışıdır.

1960'lı yıllardan itibaren bilgisayar teknolojilerindeki gelişmeler birçok mühendislik alanında olduğu gibi havacılık alanında da modern analiz yöntemlerinin hızla gelişmesine yol açmıştır. Bilgisayarlardaki gelişmelere paralel olarak nümerik analiz alanındaki gelişmelerin de desteğiyle havacılığın aerodinamik, yapı ve kontrol gibi temel alanlarında analiz çalışmaları daha kısa sürede ve daha hassas sonuçlar verecek biçimde gerçekleştirilir hale gelmiştir.

Analiz yöntemlerindeki bu gelişmenin yanında bir başka alanda, eniyileme (optimizasyon) alanındaki gelişmeler de uçak tasarımının daha kısa zamanda ve daha güvenilir biçimde yapılmasına ayrı bir katkı sağlamıştır. İlk bakışta bir uçağı tanımlayan parametrelerden, uçaktan beklenen performansları sağlayacak en uygun olanlarının seçilmesi için, söz konusu parametrelerin sistematik olarak değiştirilmesi ve her bir kombinasyon için analizlerin tekrarlanması mümkün gibi gözükse de bunun yine olağanüstü biçimde zaman alacağı ve maliyetinin katlanılamayacak kadar yüksek olacağı kolaylıkla tahmin edilebilir. Eniyileme teknikleri işte bu noktada devreye girerek parametrelerin uygun seçimini daha kestirme yollardan yapmaya imkan sağlamaktadır. Bu da parametrelerin daha az sayıda kombinasyonu için analizler yapılması ve dolayısıyla zaman ve kaynak tasarrufu anlamına gelmektedir.

Sunulan tez çalışmasının amacı bu çerçevede uçakların kanat-kuyruk kombinasyonları için aerodinamik tasarım eniyilemesi yapmak üzere bir bilgisayar yazılımının geliştirilmesidir.

1.2 Literatür Araştırması

Tez çalışmasında analiz yöntemi olarak, bilgisayar için programlanması nispeten daha kolay olan ve hızlı sonuç veren Girdap Kafes Yöntemi tercih edilmiş, eniyileme yöntemi ise genetik algoritma esaslı olarak seçilmiştir. Bu bakımdan, önce bu iki alanda bir literatür incelemesi yapılmıştır.

1.2.1 Girdap Kafes Yöntemi

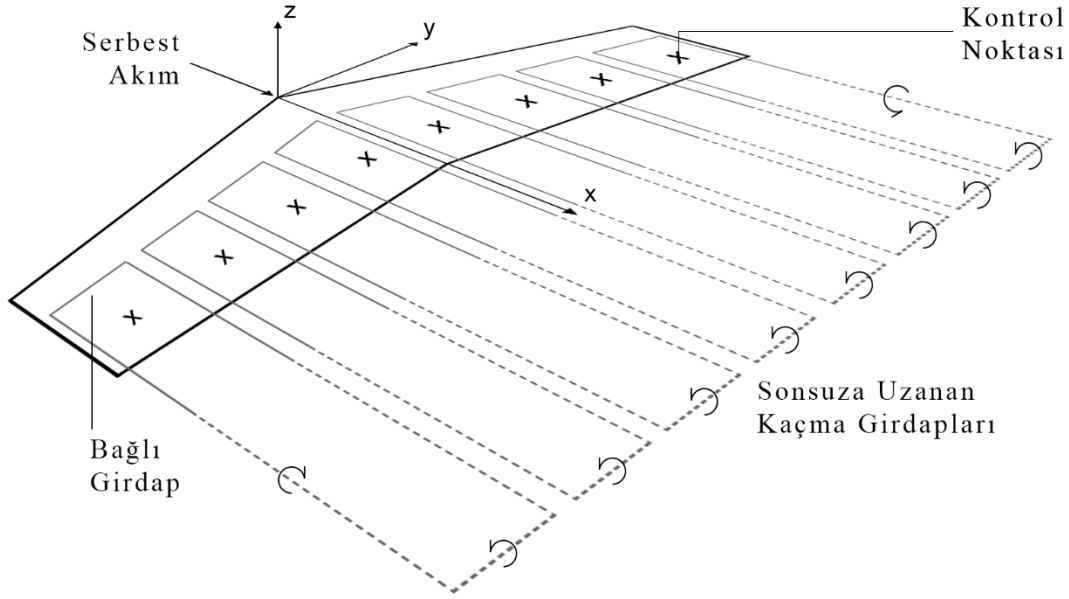
Günümüz teknolojik imkânları çerçevesinde bir uçağın aerodinamik analizleri için uygulanabilecek en üst düzeydeki teknik Navier-Stokes denklemlerinin nümerik çözümlemesini yapan Hesaplama Akışkanlar Dinamiği (CFD) yöntemleridir. Bu yöntemlerin uygulanabilmesi için öncelikle analizi yapılacak geometri etrafında yeterince geniş bir bölgede bir ağ yapısı oluşturulması gerekmektedir. Bu ağ yapısının oluşturulması zahmetli ve zaman alıcı bir iş olup, ayrıca bunu gerektiği gibi yapabilmek için profesyonelce geliştirilmiş bir bilgisayar yazılımına sahip olunmalıdır. Bundan sonraki aşamada akış probleminin çözümü için de yine güvenilir bir çözücü yazılıma ihtiyaç vardır. Her iki yazılımın da kullanımının programların kiralanması ve kullanılacakları bilgisayarların temini açısından önemli bir parasal maliyeti olduğu gibi, insan gücü ve hesaplama süresi açısından da bir maliyet söz konusudur. Gerek maliyetlerin çok yüksek olması ve gerekse hesaplama sürelerinin günümüz imkânlarına rağmen yeterince kısa olmayışı, bu yöntemlerin uçak ön tasarımında kullanımını engellemektedir. Bu nedenle ön-tasarım çalışmalarında daha düşük düzeydeki aerodinamik analiz yöntemleri tercih edilmektedir.

Taşıyıcı yüzeylerin aerodinamik analizlerine yönelik olarak geliştirilen yöntemlerin tarihçesine göz atıldığında, bu alandaki ilk çalışmaların Prandtl'ın taşıyıcı çizgi modeline dayandığı görülmektedir. Ludwig Prandtl ve meslektaşları Göttingen, Almanya'da 1912-1922 yılları arasında çeşitli çalışmalar yürütmüştür. Prandtl, bu çalışmalar sırasında 1918 yılında taşıyıcı çizgi (PLL) teorisini ortaya koymuş, ancak, o zamanlarda problemin genel analitik çözümünü tam olarak açıklayamamıştır (Anderson, 2016). Zaman içerisinde birçok araştırmacının katkılarıyla uygulaması geliştirilen bu model günümüzde dahi uçak ön tasarım çalışmalarında başarıyla kullanılmaktadır.

PLL yönteminde taşıyıcı yüzeyler iç içe geçmiş sonsuz sayıda atnalı girdabı şeklinde tanımlanır (Şekil 1.1). Kaçma girdaplarının kanat üzerinde yarattığı aşağı sapma etkisi

kanattan oluşan taşıyıcı yüzey sistemlerinin incelenmesi PLL ile mümkün olamamaktadır (Katz ve Plotkin, 2001).

PLL yönteminin yukarıda belirtilen eksikliklerini gidermek üzere geliştirilen bir sayısal çözüm yöntemi Nümerik Taşıyıcı Çizgi (NLL) Yöntemi olarak bilinmektedir. NLL yönteminde, PLL yönteminden farklı olarak atnalı girdapları sonsuz sayıda ve iç içe değil de, sonlu sayıda ve yanyana yerleştirilmektedir (Şekil 1.2).



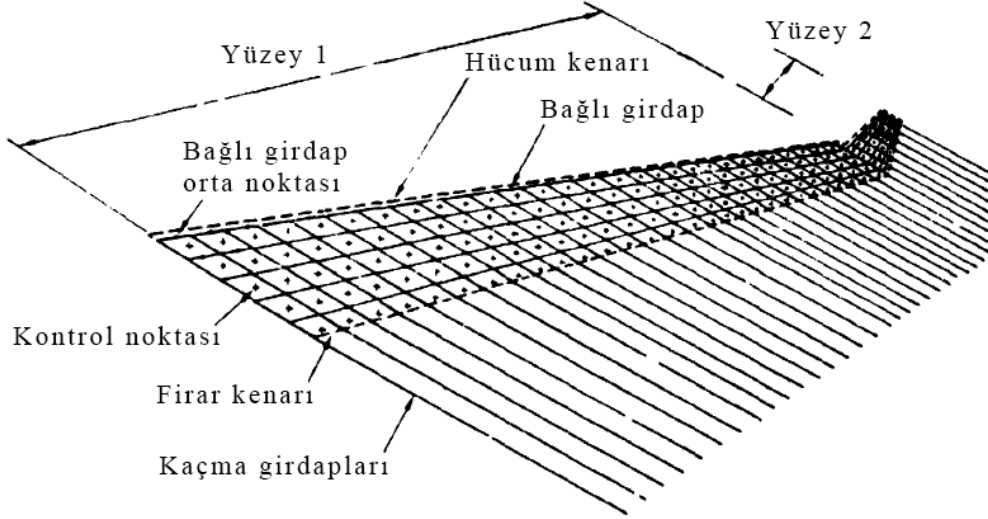
Şekil 1.2 : Nümerik taşıyıcı çizgi modeli.

Bu yöntemde taşıyıcı yüzeyler sadece açıklık doğrultusunda bir sıra atnalı girdapı ile temsil edildiğinden yalnızca simetrik profilli kanatların incelenmesinde kullanılabilir. Kambur profilli kanatlar için daha kapsamlı bir yöntem uygulanması gerekmektedir.

Girdap-kafes terimi ilk kez V. Falkner tarafından 1947 yılında kullanılmıştır ve NLL yöntemindeki yaklaşıma ek olarak veter uzunluğu boyunca da atnalı girdaplarının yerleştirilmesi prensibini esas alır (Şekil 1.3).

1950'li yıllara kadar VLM üzerine yoğun çalışmalar yürütülmüş olsa da, zamanın bilgisayarları için kapasite bakımından fazla talepkâr bir yöntem olmuştur. Girdap-kafes üzerine yürütülen çalışmaların yaygınlaşması 1960'lı yılları bulmuştur. 1965 yılı sayısal Girdap Kafes Yönteminin başlangıç yılı sayılır. Bu tarihten itibaren, bilgisayar teknolojisinin de hızla gelişmesiyle birlikte, sayısal çalışmaların sayısı katbekat artmıştır. 1976 yılında, Amerika Birleşik Devletleri'nin Virginia eyaletindeki Hampton şehrinde, Amerikan Ulusal Havacılık ve Uzay Dairesi'ne (NASA) bağlı Langley

Araştırma Merkezinde gerçekleştirilen “Girdap Kafes Kullanımı” başlıklı çalışmada, yöntemin tarihçesi ve kullanımı hakkında birçok bilgi paylaşılmıştır (DeYoung, 1976).



Şekil 1.3 : Çok elemanlı kanat yüzeyi için girdap-kafes modeli (DeYoung, 1976).

1.2.2 Genetik Algoritma

Genetik algoritmalar stokastik arama ve eniyileme algoritmalarından olan evrimsel algoritmaların (EA) bir alt sınıfıdır (Sivanandam ve Deepa, 2007). Temelleri J. H. Holland, meslektaşları ve öğrencileri tarafından 1975 yılında, Michigan Üniversitesi'nde atılmıştır. Michigan'da yürütülen araştırmaların 2 odak noktası vardır: doğal sistemlerin uyumsal süreçlerini açıklamak ve doğanın bu mekanizmasını uygulayan yapay sistemleri tasarlamak. Bu yaklaşım hem doğal hem de yapay sistemler biliminde yeni bulguların ortaya çıkmasını sağlamıştır (Goldberg, 1989).

Genetik algoritmaların temel özelliği dayanıklı olmalarıdır. Yapay sistemlerde dayanıklılığın gerekliliği açıktır. Eğer yapay bir sistem daha dayanıklı yapılırsa, maliyetli yeniden tasarım süreçleri kısaltılır ya da tamamen giderilir. Ve eğer daha yüksek bir uyumluluk seviyesine ulaşırsa, mevcut sistemler fonksiyonlarını daha uzun ve daha etkin bir şekilde gerçekleştirebilirler. Kendi kendini onarma, geliştirme ve etkinleştirme özellikleri biyolojik sistemlerin kuralıdır, ancak en gelişmiş yapay sistemlerde dahi bu özelliklere rastlamak çok zordur (Goldberg, 1989).

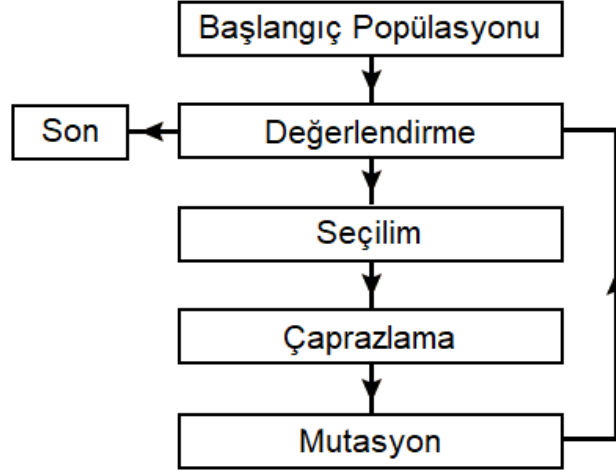
Genetik algoritma doğanın genetik mekanizmalarından ilham alır (Chong ve Zak, 2013). Darvinci evrim teorisindeki (DET) “en güçlünün hayatta kalması” (survival of the fittest) prensibini yapılı ancak rastgele bilgi alışverişi ile yaratıcı bir şekilde

birleřtirir. Her nesilde, yeni bir yapay birey topluluęu, eski ve güçlü bireylerin parçalarından meydana getirilir. Bu yeni bireyler, daha da güçlü bireyleri aramak için oluşturulmuş arama noktalarıdır.

Her birey atalarının özelliklerini ve genlerini kendinde barındırır. Darvinci Evrim Teorisinde bu genlerin nesilden nesile nasıl taşındığı anlatılmaktadır ve bu süreç 3 temel prensiple açıklanır; kalıtım (heredity), deęişim (variance) ve seçilim (selection) (Shiffman, 2012):

1. Kalıtım: Bir nesle ait bireylerin, özelliklerini sonraki nesillere aktardığı bir işlem olmalıdır. Bu özellikler bireyler üreyecek kadar uzun yaşayabilirse ancak yeni nesle ait bireylere aktarılır.
2. Deęişim: Bir nesle ait bireylerin kalıtımsal özellikleri deęişkenlik göstermelidir veya deęişime mahal verecek bir mekanizma bulunmalıdır. Örneğin, bir topluluktaki her bireyin kalıtımsal özellikleri birbirinin aynısı olursa, deęişim gerçekleşmediği sürece bu toplulukta doğan bireyler birbirlerinin veya ebeveynlerinin aynısı olacaktır. Bu da, yeni özellik kombinasyonları oluşmayacağından dolayı evrimden söz edilemeyeceği anlamına gelir.
3. Seçilim: Bir nesildeki bireylerden bazılarının ebeveyn olmasını, bazılarının ise olamamasını sağlayan bir mekanizma olmalıdır. Bu mekanizmaya genellikle “doęal seçilim” denir. Örneğin, bir popülasyona ait bireyler hayatları boyunca belli bir zorlukla mücadele ediyorlar diye düşünürsek, bu mücadeleye en çok dayanım gösteren bireylerin hayatta kalması ve sonuç olarak ebeveyn olup genlerini bir sonraki nesle aktarması zayıf bireylere nazaran çok daha muhtemeldir. Doğadaki bu mekanizma güçlü bireylerin genlerini sonraki nesillere aktarmasını ve yeni nesillerin eski nesillere göre daha dayanıklı olabildiğini sağlamaktadır.

Yukarıda belirtilen temel ilkelerden yola çıkılarak oluşturulan genetik algoritma geliştirilen bilgisayar programına şekil verir (Şekil 1.4). Oluşturulan ilk popülasyon belirlenmiş sınırlar dahilinde rastgele kalıtımsal özelliklere sahip bireylerle doldurulur. Her nesle ait bireyin uygunluğu (fitness) hesaplanarak güçlü bireyler seçilir. Seçilen bireyler çaprazlama ve mutasyon süreçlerinden sonra özelliklerini sonraki nesle aktarır. Temel ilkeler sağlandığı sürece nesil devam eder.



Şekil 1.4 : Genetik algoritmalar için genel akış çizelgesi.

Genetik algoritmaları diğer arama ve eniyileme işlemlerinden ayıran en önemli özellikler şu şekilde sıralanabilir:

1. GA değişkenlerin kendileri yerine değişkenlerin programlanması ile ilgilenir,
2. GA bir nokta yerine noktalar topluluğu ile ilgilenir,
3. GA maksatlı bilgiyi kullanır, türev veya yardımcı özelliklerle ilgilenmez,
4. GA gerekirci kurallar yerine olasılıksal bağlantı kurallarını uygular.

Verimli ve etkili bir arama sunan genetik algoritmaların kullanım alanı her geçen gün daha da artmaktadır. Bunun sebebi GAların kolay programlanabilir ve kolay düzenlenebilir olmalarıdır. Ayrıca, diğer arama algoritmaları gibi tasarım uzayındaki türevin varlığı, süreklilik vb. kısıtlayıcı kabullerden temelde yoksundur (Goldberg, 1989).

GAların ne kadar kullanışlı olduğunu açıklamak üzere şöyle bir düşünce deneyi yapılabilir: bir maymun klavye başına geçip rastgele tuşlara basarak Shakespeare'in Hamlet isimli eserini yazabilir mi? Bu deneye "sonsuz maymun deneyi" denir. Bütün kitap yerine, kitaptan sadece bir cümle alalım: "olmak ya da olmamak işte bütün mesele bu". Daha da basitleştirmek için noktalama kullanmadan, sadece küçük harfler kullanalım. Bu cümle boşluklar da dâhil 40 karakterden oluşmaktadır. Türkçe bir klavyede toplam 32 adet harfe ilave bir de boşluk tuşu vardır. Maymun sadece bu 33 tuşu kullanacak olursa, 40 karakterli bir cümleyi rastgele yazabilme olasılığı $(1/33)^{40}$ kadar hesaplanır. Bu da demek oluyor ki, maymunun cümleyi yazabilme olasılığı yaklaşık 5.5×10^{60} 'da 1 kadar olur. Bu, kitap içeriğindeki sadece bir cümle

için hesaplanan olasılıktır. Kısacası, her bir deneme 1 saniye sürse bile, yaklaşık 1.4×10^{10} yaşında (4.4×10^{17} saniye) olduğu düşünülen evrende kitaptan bir cümleyi bile yazabilmenin imkansız olduğu anlaşılır.

D. Shiffman sonsuz maymun deneyini bir değil, birçok bireyden oluşan maymun popülasyonu üzerinde GA aracılığıyla yapmıştır (Shiffman, 2012). Kodlamasında, nesiller sırasında doğru karakterleri tutturabilen sanal maymunlara yüksek uygunluk değeri veren Shiffman, kitabın İngilizce basımındaki 39 karakterden oluşan “to be or not to be that is the question” cümlesini dakikalar içerisinde yazdırabilmiştir (Shiffman, 2017). Biraz daha kısa hali olan “to be or not to be” cümlesini saniyeler içerisinde yazdırabilmek dahi mümkün olmaktadır (Chang, 2018).

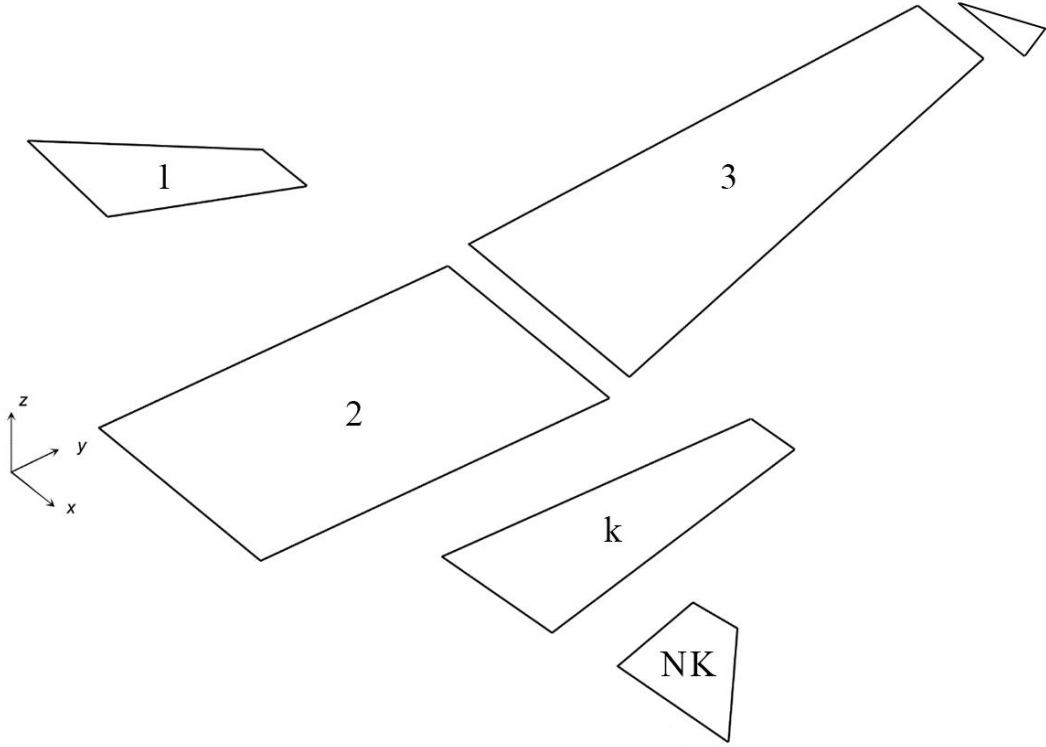
Sonuç olarak insanoğlu, tasarım yaparken ilham aldığı doğayı ve kanunlarını yine tasarlanan sistemin eniyilemesi sürecine katarak, en uygun sistemi hızlı ve verimli bir şekilde geliştirebilmektedir.

2. GİRDAP KAFES YÖNTEMİ

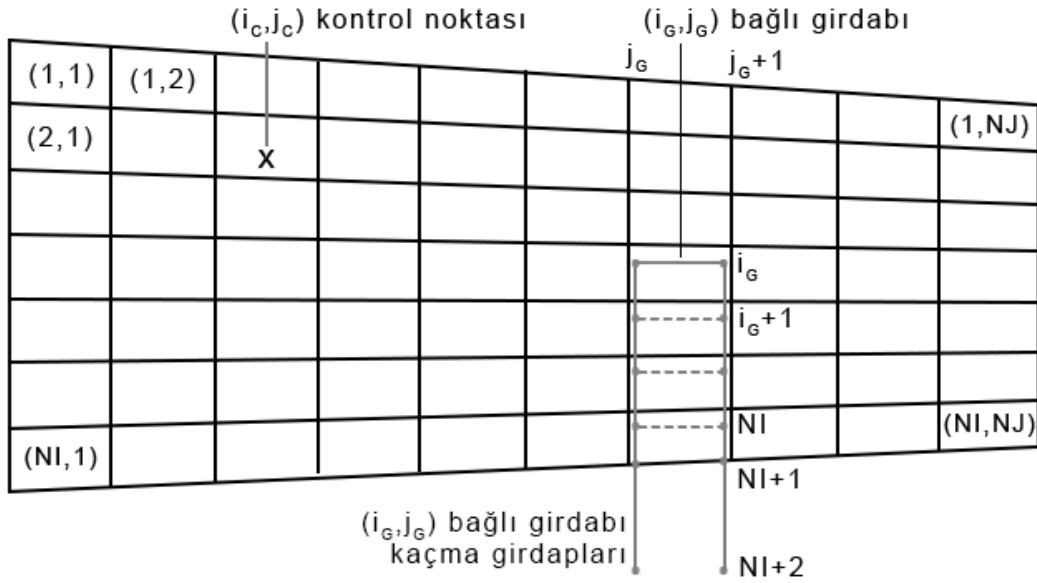
Bu bölümde tez çalışmasındaki eniyileme yönteminde analizler için kullanılan girdap kafes yönteminin esasları ve formülasyonu izah edilmekte olup bu yöntemin uygulamasına yönelik olarak geliştirilen bilgisayar programının doğrulanması için yapılan çeşitli uygulamalarına yer verilmektedir.

2.1 Çok Elemanlı Taşıyıcı Yüzey Sistemleri İçin Girdap Kafes Yöntemi

Herhangi bir taşıyıcı yüzey sistemi üzerinde VLM yönteminin uygulanması öncesinde, sistem Şekil 2.1'deki gösterildiği gibi elemanlarına ayrılır. Bu yüzeylerin her biri de Şekil 2.2'de gösterildiği gibi açıklık ve veter doğrultusunda panellere bölünür. Daha sonra her bir panel üzerine birer atnalı girdabı yerleştirilerek aerodinamik akış problemi serbest akım içerisinde yer alan bu atnalı girdapları sistemi ile temsil edilir.

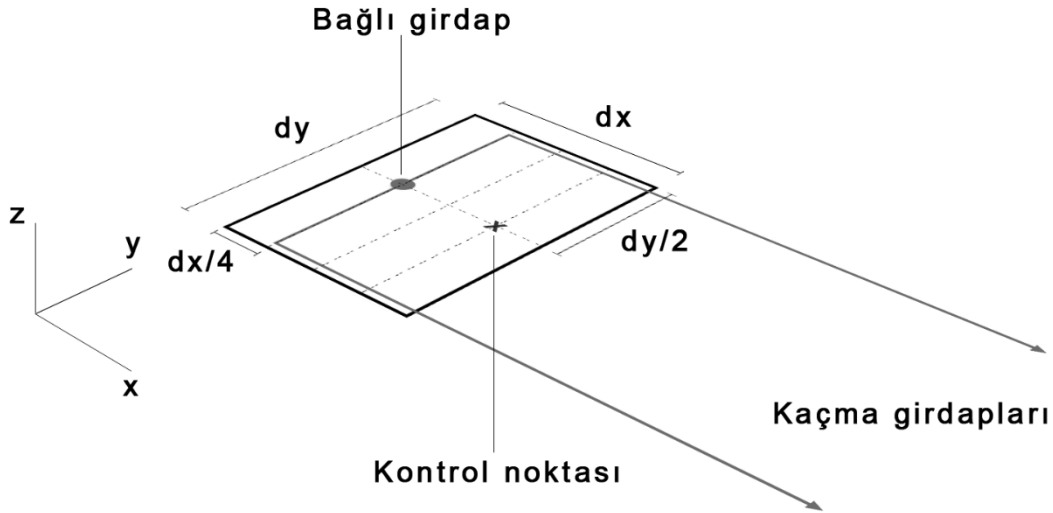


Şekil 2.1 : Çok elemanlı taşıyıcı yüzey sistemi.



Şekil 2.2 : Taşıyıcı yüzeyin açıklık ve veter doğrultusunda panellere ayrılması ve atnalı girdaplarının yerleşimi.

Herbir panel üzerindeki atnalı girdabı, panelin çeyrek veter çizgisi üzerinde yer alan bir bağlı girdap ile bağlı girdabın iki ucundan çıkarak panel kenarı doğrultusunda firar kenarına kadar yüzeyi izleyen, firar kenarından sonra ise serbest akım doğrultusunda akım gerisine yönelerek sonsuz giden iki kaçma girdabından oluşmaktadır (Şekil 2.3).

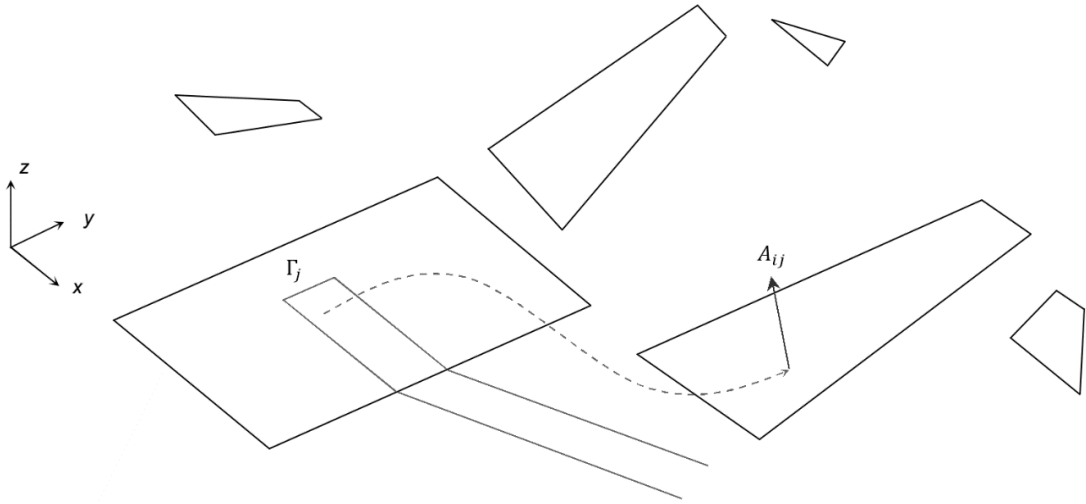


Şekil 2.3 : Çeyrek-üççeyrek kuralına göre panel üzerinde atnalı girdabı ve kontrol noktası yerleşimi.

Serbest akım içerisinde yer alan taşıyıcı yüzey sistemi bu şekilde atnalı girdaplarıyla temsil edilmekle birlikte, bu atnalı girdaplarının şiddetleri problemin bilinmeyenleridir. Girdap şiddetlerinin hesaplanması için yüzey üzerindeki sınır koşulundan yararlanır. Sınır koşulu akımın yüzeye teğet olmasını gerektirir ki çoğu

potansiyel akım yönteminde bu koşul akımın yüzeye dik hız bileşeninin sıfır olması şeklinde uygulanır. Girdap kafes yönteminde de herhangi bir j paneli üzerindeki Γ_j şiddetine sahip atnalı girdabının herhangi bir i paneli üzerinde belirtilmiş olan bir kontrol noktasında yarattığı dikey hız $A_{ij}\Gamma_j$ olmak üzere i kontrol noktasında tüm panellerin yarattığı hızlarla serbest akımın dikey hız bileşeni toplanarak sıfıra eşitlenmek suretiyle bir denklem elde etmek mümkündür (Şekil 2.4). Her bir kontrol noktası için benzeri işlem yapılarak sonuçta panel sayısınca veya diğer bir deyişle bilinmeyen girdap şiddetleri sayısınca denklemden oluşan bir denklem takımı elde edilir:

$$\sum_{j=1}^N A_{ij}\Gamma_j = \vec{V}_\infty \cdot \vec{n}_i \quad (i = 1, 2, \dots, N) \quad (2.1)$$



Şekil 2.4 : Bir atnalı girdabının bir kontrol noktasındaki etkisi.

(2.1) denklem sisteminde kullanılan indisleme bu haliyle girdap kafes yönteminin tek bir taşıyıcı yüzey için ve sadece açıklık boyunca bir sıra panel olması halinde (NLL) bilgisayarla uygulanmasına uygundur. Sistemde birden fazla taşıyıcı yüzey olduğunda ve veter doğrultusunda panelleme yapıldığında (VLM) farklı bir düzenleme gerekir. Buradaki uygulamalarda VLM için Şekil 2.2’de olduğu gibi bir indisleme uygulanmakta olup ayrıca çoklu yüzeyler için bu indislere bir üçüncü indis, taşıyıcı yüzeyin temsilcisi olarak eklenmektedir.

(2.1) denklem sistemindeki indisleme tek bir taşıyıcı yüzeyin analizi için uygun olmakla birlikte, içerisinde kanat, kanatçık, yatay kuyruk, düşey kuyruk gibi birden fazla taşıyıcı yüzeyin olduğu bir sistem için pek uygun değildir. Analizin bilgisayar

programına daha uygun hale getirilmesi için burada aşağıdaki gibi bir indisleme tercih edilmiştir. Denklem sistemi bu yeni indisleme ile

$$\sum_{k_G=1}^{NK} \sum_{i_G=1}^{NI} \sum_{j_G=1}^{NJ} A_{(i_C, j_C, k_C)}^{(i_G, j_G, k_G)} \Gamma^{(i_G, j_G, k_G)} = R_{(i_C, j_C, k_C)} \quad (2.2)$$

$$\left(\begin{array}{l} i_C = 1, 2, \dots, NI_{k_C} \\ j_C = 1, 2, \dots, NJ_{k_C} \end{array} \right) (k_C = 1, 2, \dots, NK)$$

şekline gelmektedir. Buradaki indislerin altında yer alan C bu indisin kontrol noktasına ait olduğunu, G ise atnalı girdabına ait olduğunu işaretlemek üzere kullanılmaktadır. Kontrol noktası indislemesi alt-indis olarak, girdaplara ait indislemeler ise üst-indis olarak gösterilmiştir. Buna göre $\Gamma^{(i_G, j_G, k_G)}$ büyüklüğü k_G indisli taşıyıcı yüzey üzerindeki (i_G, j_G, k_G) indisli atnalı girdabının şiddetini belirtmektedir. $A_{(i_C, j_C, k_C)}^{(i_G, j_G, k_G)}$ büyüklüğü ise (i_G, j_G, k_G) girdabının (i_C, j_C, k_C) kontrol noktasında birim girdap şiddeti için indüklediği hızın dik bileşenidir:

$$A_{(i_C, j_C, k_C)}^{(i_G, j_G, k_G)} = \vec{v}_{(i_C, j_C, k_C)}^{(i_G, j_G, k_G)} \cdot \vec{n}_{(i_C, j_C, k_C)} \quad (2.3)$$

(2.2) Denklem sistemi

$$[A_{ij}] \cdot \{\Gamma_j\} = \{R_j\} \quad (2.4)$$

gibi matris formda ifade edilebilir. Burada

$$[A_{ij}] = \begin{bmatrix} A_{(1,1,1)}^{(1,1,1)} & \dots & A_{(1,NJ_1,1)}^{(1,1,1)} & \dots & \dots & A_{(i_C, j_C, 1)}^{(i_G, j_G, 1)} \\ \vdots & \ddots & \vdots & \dots & \dots & \vdots \\ A_{(NI_1, 1, 1)}^{(1,1,1)} & \dots & A_{(NI_1, NJ_1, 1)}^{(1,1,1)} & \dots & \dots & A_{(i_C, j_C, NK)}^{(i_G, j_G, 1)} \\ \vdots & \vdots & \vdots & \dots & \dots & \vdots \\ \dots & \dots & \dots & \dots & \dots & \vdots \\ A_{(i_C, j_C, 1)}^{(i_G, j_G, NK)} & \dots & \dots & A_{(1, 1, NK)}^{(NI_{NK}, NJ_{NK}, NK)} & \dots & A_{(1, NJ_{NK}, NK)}^{(NI_{NK}, NJ_{NK}, NK)} \\ \dots & \dots & \dots & \vdots & \dots & \vdots \\ \dots & \dots & \dots & A_{(NI_{NK}, NJ_{NK}, NK)}^{(NI_{NK}, NJ_{NK}, NK)} & \dots & A_{(NI_{NK}, NJ_{NK}, NK)}^{(NI_{NK}, NJ_{NK}, NK)} \\ \dots & \dots & \dots & \vdots & \dots & \vdots \\ \dots & \dots & \dots & A_{(NI_{NK}, 1, NK)}^{(NI_{NK}, NJ_{NK}, NK)} & \dots & A_{(NI_{NK}, NJ_{NK}, NK)}^{(NI_{NK}, NJ_{NK}, NK)} \end{bmatrix} \quad (2.5a)$$

$$\{\Gamma_j\} = \begin{bmatrix} \Gamma^{(1,1,1)} \\ \vdots \\ \Gamma^{(NI_1, NJ_1, 1)} \\ \vdots \\ \Gamma^{(1,1, NK)} \\ \vdots \\ \Gamma^{(NI_{NK}, NJ_{NK}, NK)} \end{bmatrix} \quad (2.5b)$$

$$\{R_j\} = \begin{bmatrix} R_{(1,1,1)} \\ \vdots \\ R_{(N1, N1, 1)} \\ \vdots \\ R_{(1,1, NK)} \\ \vdots \\ R_{(N1, NK, N1, NK)} \end{bmatrix} \quad (2.5c)$$

ve ayrıca,

$$R_{(i_c, j_c, k_c)} = \vec{V}_\infty \cdot \vec{n}_{(i_c, j_c, k_c)} \quad (2.6)$$

dir. (2.5) denklem sistemi, tüm katsayıları hesaplandıktan sonra Gauss eliminasyon veya benzeri bir yöntemle çözülerek atnalı girdaplarının şiddetleri elde edilir. Katsayıların hesaplanmasına ilişkin ayrıntılara izleyen bölümde yer verilmektedir.

2.2 Aerodinamik Katsayıların Hesaplanması

Taşıyıcı yüzeyin aerodinamik etkilerini sadece her bir panel üzerindeki bağlı girdapların yarattığı kabul edilmektedir. Buna göre herhangi bir (i, j, k) panelinin üzerinde yaratılan aerodinamik kuvvet ve moment

$$\vec{f}_{ijk} = \rho_\infty \vec{V}_{e_{ijk}} \times \Gamma_{ijk} \Delta \vec{s}_{ijk} \quad (2.7)$$

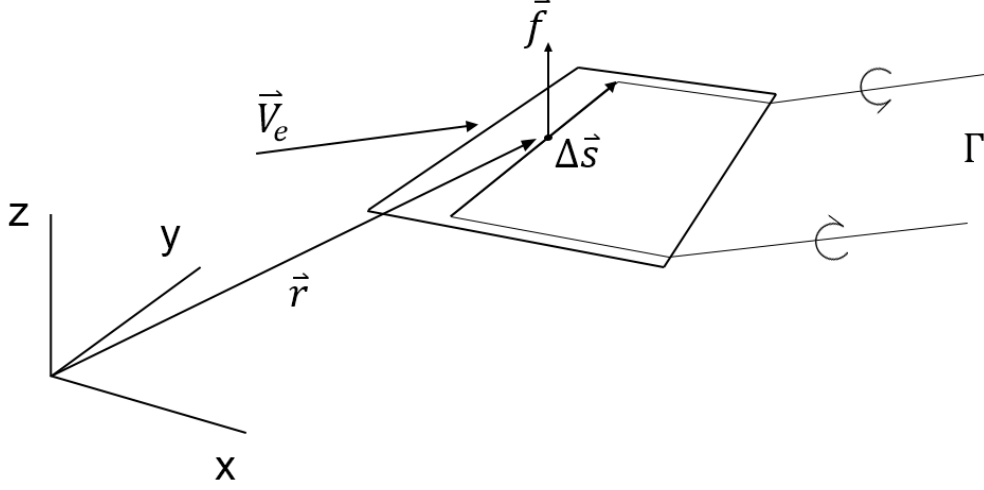
$$\vec{m}_{ijk} = \vec{r}_{ijk} \times \vec{f}_{ijk} \quad (2.8)$$

şeklinde hesaplanır. (2.7) denklemi Joukowski taşıma kanunun vektörel bir ifadesi olup $\vec{V}_{e_{ijk}}$ vektörü (i, j, k) paneli üzerindeki etkin hızdır. $\Gamma_{ijk} \Delta \vec{s}_{ijk}$ vektörü ise panel üzerindeki bağlı girdap vektörüdür. (2.8) denklemindeki \vec{r}_{ijk} vektörü bağlı girdabın orta noktasının konum vektörüdür (Şekil 2.5).

Herhangi bir panele etkiyen lokal aerodinamik kuvvetin tanımlandığı (2.7) bağıntısındaki etkin hız serbest akım hızı ile tüm girdap sisteminin o panele ait bağlı girdabın orta noktasında indüklediği bileşke hızın toplamına eşittir:

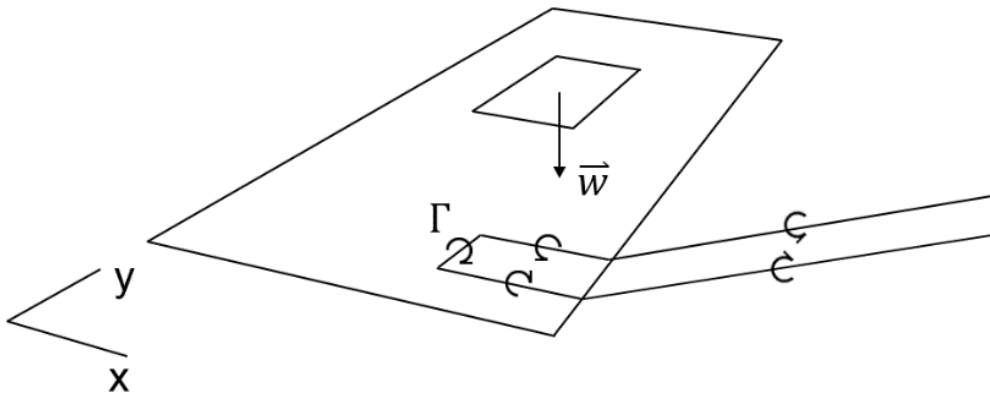
$$\vec{V}_{e_{(i_c, j_c, k_c)}} = \vec{V}_\infty + \vec{w}_{(i_c, j_c, k_c)} \quad (2.9)$$

$$\vec{w}_{(i_c, j_c, k_c)} = \sum_{k_G=1}^{NK} \sum_{i_G=1}^{NI_{k_G}} \sum_{j_G=1}^{NJ_{k_G}} \Gamma^{(i_G, j_G, k_G)} \vec{v}_{(i_c, j_c, k_c)}^{(i_G, j_G, k_G)} \quad (2.10)$$



Şekil 2.5 : Herhangi bir panel üzerindeki etkin hız ve aerodinamik kuvvet vektörleri.

(2.9) denkleminde belirtilen $\vec{w}_{(i_c, j_c, k_c)}$ vektörü, (i_c, j_c, k_c) paneli üzerindeki toplam aşağı sarmayı ifade etmektedir (Şekil 2.6). Bu büyüklüğün hesabı için hesaplama da dekle (2.10)'da gösterildiği gibidir. Burada $\vec{v}_{(i_c, j_c, k_c)}^{(i_G, j_G, k_G)}$ büyüklüğü herhangi bir (i_G, j_G, k_G) paneline ait atnalı girdabının (i_c, j_c, k_c) paneline ait bağlı girdabın orta noktasında birim şiddet başına indüklediği hızı temsil etmektedir.



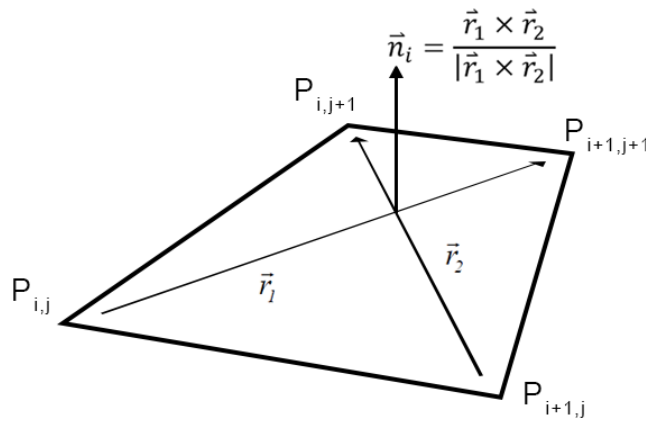
Şekil 2.6 : Herhangi bir panel üzerinde oluşan aşağı sapma.

Her bir panel üzerinde yaratılan kuvvet ve momentler tüm taşıyıcı sistem için toplanarak

Yukarıda verilen aerodinamik katsayı hesaplamalarında q_∞ dinamik basıncı, S_{ref} ise referans kanat alanını ifade eder.

2.3 Denklem Takımındaki Katsayıların Hesaplanması

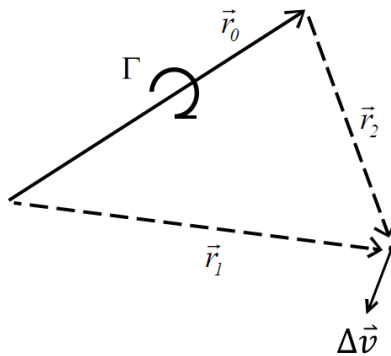
(2.4) denkleminde A matrisi panellerin kontrol noktalarında birim girdap şiddeti için indüklenen hızlara tekabül ederken, R matrisi serbest akımın etkisini temsil etmektedir. Köşe noktaları yardımıyla çizilen köşegenlerin vektörel çarpımının normuna bölümü (\vec{n}_i) yerel normal vektörü verecektir (Şekil 2.8).



Şekil 2.8 : Panel normali için köşegen çarpımı gösterimi.

Birim şiddet için indüklenen hızlar da Biot-Savart kanunu ile hesaplanır (Şekil 2.9). Bir atnalı girdabının bağlı girdabı ve kaçma kolları ayrı ayrı düşünülerek, her bir girdap çizgisinin indüklediği hız vektörü aşağıdaki gibi hesaplanır:

$$\Delta \vec{v} = \frac{\Gamma}{4\pi} \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|^2} \vec{r}_0 \cdot \left(\frac{\vec{r}_1}{|\vec{r}_1|} - \frac{\vec{r}_2}{|\vec{r}_2|} \right) \quad (2.18)$$



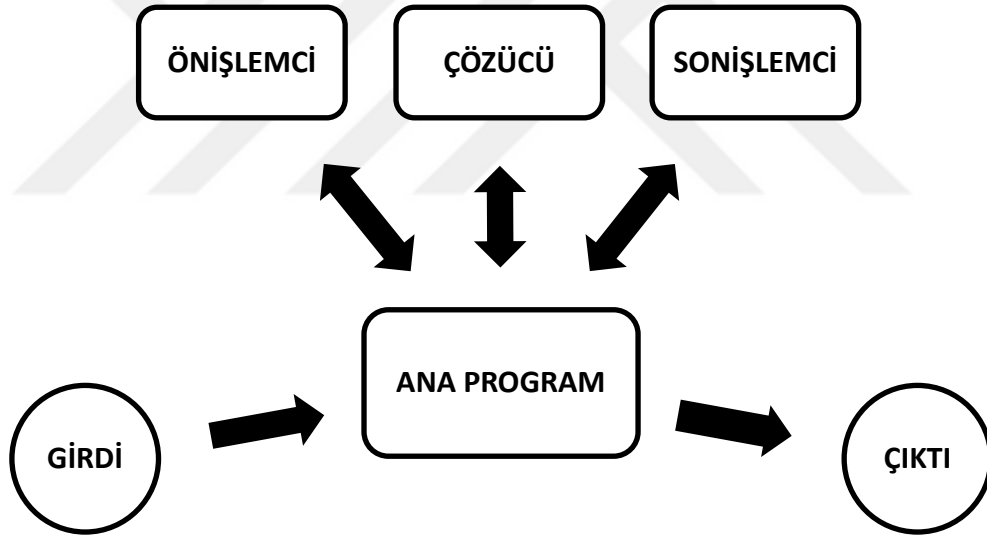
Şekil 2.9 : Kontrol noktasında indüklenen hız vektörü (Biot-Savart Kanunu).

Denklem 2.18'deki gibi hesaplanan indüklenmiş hızlar, panel yerel normalleri ile (2.3) denklemindeki gibi vektörel çarpılarak katsayılar matrisi oluşturulur.

2.4 Programlama

Bu kısımda Girdap Kafes Yöntemini çok elemanlı herhangi bir taşıyıcı yüzey sistemi için uygulayan jenerik bir bilgisayar programının yazım biçimi anlatılmaktadır. Kolay kullanılabilir olması ve kullanıcıya birçok iç fonksiyon sunması dolayısıyla MATLAB programlama dili kullanılmıştır.

Ön işleme, çözüm ve son işleme aşamalarını yürütecek olan 3 alt-program yazılmıştır ve yine bu programların altında çağırılan alt programlar bulunmaktadır. Bilgisayar programı yazılan bir ana program üzerinden yürütülmektedir ve girdiler de bu programa girilmektedir (Şekil 2.10).



Şekil 2.10 : VLM programı genel akış çizeneği.

Ana programa girilmesi gereken girdiler aşağıdaki gibidir:

- Serbest akım tanımlamaları
 - U : hız (m/s)
 - h : irtifa (m)
 - α : geliş açısı ($^\circ$)
- Taşıyıcı yüzey tanımlamaları
 - *profil* : yüzey kesit profili
 - S : planform alan (m^2)
 - AR : açıklık oranı

- λ : sivrilme oranı
- α_{in} : oturma açısı (°)
- α_{tw} : burulma açısı (°)
- α_{sw} : ok açısı (°)
- α_{di} : dihedral açısı (°)
- H : yüzey referans noktası
- R : yüzey yerleşim vektörü
- n_i : veter boyunca panel sayısı
- n_j : açıklık boyunca panel sayısı
- ss : simetri durumu (0: tek yüzey, 1: simetrik yüzey)

Referans noktası (H) ve yerleşim vektörü (R) girdileri, yüzeylerin uzayda konumlandırılmasında kullanılır. 0 ile 1 arasında seçilen H değeri, üzerinde tanımlı olduğu yüzeyin kök veterinin H noktasından tutularak R vektörü kadar ötelenir. Bu şekilde yüzey uzayda istenilen konuma yerleştirilmiş olur. Kanat ve yatay kuyruk gibi simetrik yüzeylerin tanımlanmasında ss girdisi 1 olarak girilir ise yüzey xz düzlemine göre simetrik olarak tanımlanmış olur.

Bu noktadan sonra sırasıyla önışlemci, çözücü ve sonışlemci alt-programları çağırılarak ana program ile ilgili kodlama tamamlanır. Alt-programların kodlaması ile ilgili detaylar sonraki başlıklar altında işlenmiştir.

2.4.1 Ön işleme

Hesaplamalarda kullanılacak parametrelerin oluşturulması bu aşamada gerçekleştirilir. Öncelikle serbest akımla ilgili değişkenler belirlenir. MATLAB fonksiyonlarından “*atmosisa*” tanımlanan irtifa girdisi (h) ile (2.19) denklemindeki gibi kullanılarak Uluslararası Atmosfer Standartları (ISA) değerleri çağırılır. Bu şekilde irtifadaki hava için sıcaklık (T), ses hızı (a), basınç (P) ve yoğunluk (ρ) bilgilere ulaşmak mümkündür. VLM uygulaması sırasında dinamik basınç hesabı için hava yoğunluğu bilgisi yeterli olacaktır:

$$[T, a, P, \rho] = \text{atmosisa}(h) \quad (2.19)$$

$$q_\infty = \frac{1}{2} \rho_\infty U_\infty^2 \quad (2.20)$$

Serbest akım vektörü ise aşağıdaki gibi hesaplanır:

$$V_\infty = U_\infty \begin{bmatrix} \cos \alpha \\ 0 \\ \sin \alpha \end{bmatrix} \quad (2.21)$$

Serbest akım tanımlandıktan sonra yüzeyler ile ilgili işlemlere geçilir. Tanımlı taşıyıcı yüzeyler için girdiler aşağıdaki gibi hücreler halinde kod içerisinde saklanır. Yüzeyleri bu şekilde tanımlamak genel bir yüzey tanımı yapılabilmesi açısından kolaylık sağlar.

$$yüzey\{s\} = \{profil_s, S_s, AR_s, \lambda_s, \alpha_{in_s}, \alpha_{tw_s}, \alpha_{sw_s}, \alpha_{di_s}, H_s, R_s, n_{i_s}, n_{j_s}, ss_s\} \quad (2.22)$$

Önişlemciye gönderilen yüzey girdileri işlenerek ilgili değişkenler oluşturulur. Tanımlı girdiler kullanılarak temel taşıyıcı yüzey parametreleri sırayla

$$b = \sqrt{S \cdot AR} \quad (2.23)$$

$$\bar{c} = b/AR \quad (2.24)$$

$$c_0 = \frac{3}{2} \cdot \bar{c} \cdot \frac{1 + \lambda}{1 + \lambda + \lambda^2} \quad (2.25)$$

$$c_1 = c_0 \cdot \lambda \quad (2.26)$$

şeklinde hesaplanabilir (Sadraey, 2012). (2.23) ve (2.26) denklemleri arasında hesaplanan b kanat açıklığı, \bar{c} ortalama aerodinamik veter, c_0 kök veter ve c_1 uç veter uzunluklarıdır.

Bu tez kapsamında yazılan VLM programı düz levhalara ek olarak 4 ve 6 serisi NACA profili kesitli kanatları çözebilecek şekilde yazılmıştır. 4 serisi NACA profillerinin oluşturulmasında 460 numaralı NACA raporu (Jacobs ve diğ., 1933), 6 serisi profillerin oluşturulmasında ise 4741 numaralı NASA raporundaki yönlendirmelerden yararlanılmıştır (Ladson ve diğ, 1996).

Kosinüs dağılımı ile noktaları yerleştirilmiş birim uzunluktaki veter, hücum kenarı orijin noktasına denk gelecek şekilde xy düzleminde oluşturulur. Veter üzerindeki nokta sayısı $(n_{i_s} + 1)$ kadardır. Bu birim uzunluktaki veterin üzerinde bulunan noktaların konum vektörleri (2.25) denkleminde hesaplanan kök veter uzunluğu ile çarpılarak gerçek uzunluktaki kök veteri oluşturulur. Benzer şekilde (2.26) denkleminde yararlanılarak uç veteri bulunur. Oturma ve burulma açılarının uygulanabilmesi için kök ve uç veterlerine ait noktaların konum vektörleri aşağıdaki dönme vektörleri ile çarpılır:

$$d_{kök} = \begin{bmatrix} \cos \alpha_{in} & 0 & \sin \alpha_{in} \\ 0 & 1 & 0 \\ -\sin \alpha_{in} & 0 & \cos \alpha_{in} \end{bmatrix} \quad (2.27)$$

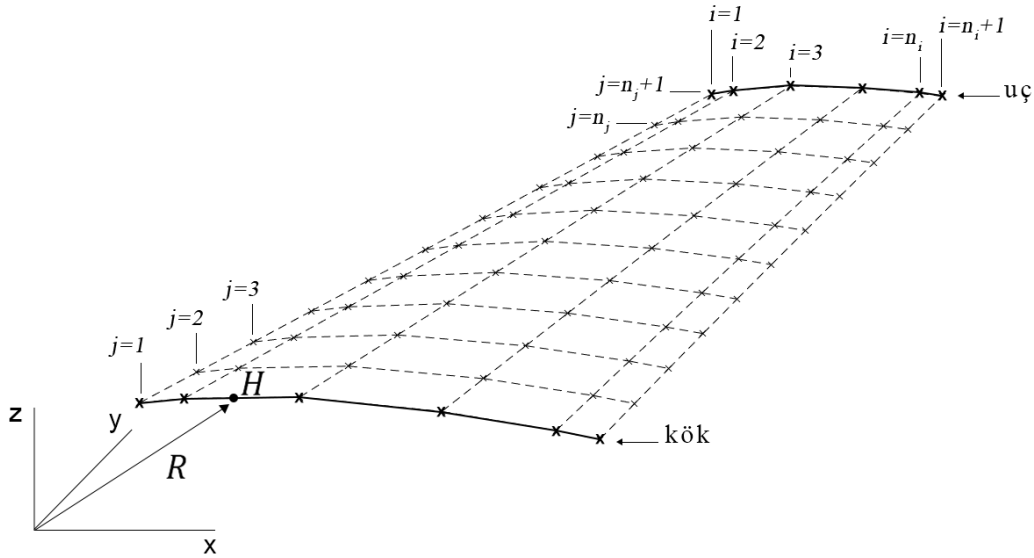
$$d_{uç} = \begin{bmatrix} \cos(\alpha_{in} + \alpha_{tw}) & 0 & \sin(\alpha_{in} + \alpha_{tw}) \\ 0 & 1 & 0 \\ -\sin(\alpha_{in} + \alpha_{tw}) & 0 & \cos(\alpha_{in} + \alpha_{tw}) \end{bmatrix} \quad (2.28)$$

Uç veteri α_{di} girdisi dikkate alınarak x eksenini etrafında aşağıdaki döndürme matrisi kullanılarak çevrilir:

$$d_{di} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_{di} & -\sin \alpha_{di} \\ 0 & \sin \alpha_{di} & \cos \alpha_{di} \end{bmatrix} \quad (2.29)$$

Bu işlemlerden sonra kök ve uç veteri arasında toplamda $(n_{j_s} + 1)$ kadar kesit olacak şekilde noktalar ile doldurulur. Ok açısının uygulanabilmesi için ise, her bir veter tanımlanan ok açısını takip edecek şekilde x ekseninde ötelenir.

Son olarak yüzey menteşe noktasından tutularak yerleşim vektörü ile hedef konuma yerleştirilir (Şekil 2.11). Oluşturulan noktalar panel köşe noktalarını temsil etmektedir.



Şekil 2.11 : Uzaydaki konumuna yerleştirilen panellere bölünmüş taşıyıcı yüzey. Oluşturulan panel köşe noktaları kullanılarak kontrol noktalarının yeri Şekil 2.3’de gösterildiği gibi hesaplanır. Panel normaleri de Şekil 2.8’de gösterildiği gibi bulunur. Bu aşamadan itibaren atnalı girdapları ile ilgili işlemlere geçilir.

Atnalı girdaplarının bağılı girdapları çeyrek-üççeyrek kuralı çerçevesinde, yine panel köşe noktalarının vektörel kullanımı ile bulunur. Kaçma girdapları, her yüzey üzerindeki bağılı girdapları veter doğrultusunda birbirine bağlayacak ve firar kenarına gelindiğinde yeteri uzunlukta tutularak serbest akımı takip eden ek kaçma girdapları oluşacak şekilde oluşturulur.

Son olarak ss girdisi 1 olan yüzeylere ait bütün noktaların xy düzlemine göre simetriği olan noktalar oluşturulur. Bunun için aşağıdaki gibi bir simetri matrisi kullanılabilir. Herhangi bir konum vektörünün bu matris ile çarpımı xz düzlemine göre simetriğini verir.

$$d_{simetri} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.30)$$

Bu noktadan itibaren çözüm için gerekli olan kontrol noktaları, normal vektörleri ve atnalı girdapları ile ilgili parametreler bulunmuş olur.

2.4.2 İşleme

Ön işleme aşamasında, yüzeylere ait kontrol noktası koordinatları, panel normal vektörleri ve atnalı girdaplarının uç noktalarının uzaydaki konumları gibi bilgiler elde edilmiştir. Bu bilgiler ana program üzerinden çözücü alt-programa iletilir. Çözücüye iletilen bu bilgiler kullanılarak denklem takımı oluşturulur.

Denklem takımı oluşturmak için yazılan bir döngü içerisinde, atnalı girdapları ve kontrol noktalarını kullanan bir diğer alt programa (VIND) iletilir. Bu alt programda sV yüzeyi üzerinde bulunan atnalı girdaplarının c kontrol noktası üzerinde birim girdap şiddeti başına indüklediği hız vektörlerini Biot-Savart Kanunu vasıtasıyla hesaplar. Ana programa iletilen vB bağılı girdap indüklemeleri, vL sol kaçma girdaplarının ve vR sağ kaçma girdaplarının indüklemeleri ile kullanılarak katsayılar matrisi oluşturulur. Aynı döngü içerisinde serbest akımın etkileri de sağ taraf matrisine yazılır.

Sonuç olarak bu içiçe döngü ile denklem takımının sol ve sağ taraf matrisleri olan A ve R matrisleri oluşturulur. Girdap şiddetlerini bulmak için denklem takımının herhangi bir cebirsel yöntem ile çözülmesi aşamasında MATLAB birçok iç fonksiyon ile çözüm sunmaktadır.

Bu tez çalışması kapsamında MATLAB yazılım paketinin LU ayrıştırma yöntemini uygulayan *linsolve* fonksiyonu kullanılmıştır. Bu fonksiyon adından anlaşılacağı gibi, lineer denklem sistemlerini çözmesi için yazılmıştır ve aşağıdaki gibi çağırılır:

$$\Gamma = \text{linsolve}(A, R) \quad (2.31)$$

2.4.3 Son işleme

Elde edilen girdap şiddetleri kullanılarak aerodinamik katsayıların hesaplaması bu aşamada yapılır. Çözücünden ana programa iletilen girdap şiddetleri, sonişlemci alt-programına yine atnalı girdaplarına ait uç nokta koordinatları ile birlikte gönderilir.

(2.7) denklemindeki gibi hesaplanan aerodinamik kuvvetlerin taşıma ve indüklenmiş sürüklenme doğrultularında izdüşürülmesiyle birlikte, yüzeye etki eden taşıma ve indüklenmiş kuvvet bileşenleri bulunmuş olur. Bağlı girdap orta noktalarının uzaydaki konumları (2.8) denklemindeki gibi kullanılarak da yüzeyler üzerindeki yunuslama momentleri bulunur. Ardından yürütülen boyutsuzlaştırma işlemleri sonucunda aerodinamik katsayılar her bir yüzey için hesaplanır.

2.5 Doğrulama Çalışmaları

VLM uygulaması için önceki başlıklar altında anlatılan bilgisayar programının doğruluğunu test etmek amacıyla önce düz bir tek-kanat için uygulama yapılarak PLL yöntemi sonuçları ile karşılaştırma yapılması öngörülmüştür. Birden fazla taşıyıcı yüzeyden oluşan bir taşıyıcı yüzey sistemi için ise PLL ile doğrudan karşılaştırma yapılması mümkün olmadığından test amacıyla bir tek-kanadın yer etkisi incelemesi göz önüne alınmıştır. Ayrıca programın değişik durumlardaki işlerliğini göstermek amacıyla kanat-kuyruk kombinasyonu için bir uygulama gerçekleştirilmiştir.

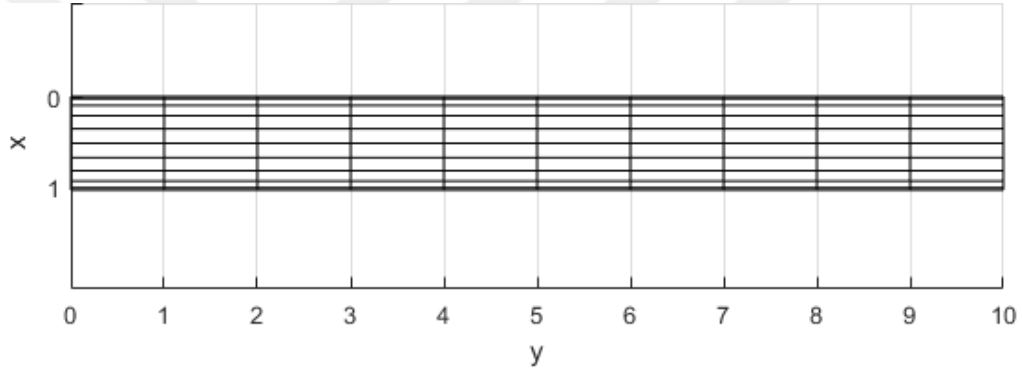
2.5.1 PLL ile karşılaştırma

Bu doğrulama analizi için dikdörtgen üstgörünümlü düz bir kanat göz önüne alınmış olup açıklık oranı 10 ve alanı 10 m² olan bir kanat VLM ve PLL yöntemleri kullanılarak analiz edilmiştir. Kesit profili olarak düz levha seçilen kanat için girdiler Çizelge 2.1’de görüldüğü gibidir.

Çizelge 2.1 : Tek-kanat için doğrulama testinde PLL uygulaması için girdiler.

Girdi	Sembol	Değer
Kanat alanı	S_{ref}	$10 m^2$
Açıklık oranı	AR	10.0
Sivrilme oranı	λ	1.0
Geliş açısı	α_{in}	$0^\circ - 5^\circ$
Burulma açısı	α_{tw}	0°
Sıfır taşıma açısı	α_0	0°
Taşıma eğrisi eğimi	$C_{l\alpha_{max}}$	2π

PLL yöntemi uygulanırken kanat açıklığı boyunca 20 istasyon kullanılmıştır. VLM uygulamasında ise kanat, açıklığı boyunca eşit genişlikte olmak üzere ve veter boyunca kosinüs dağılımı ile 10ar panelden toplam 100 panele bölünmüştür (Şekil 2.12).



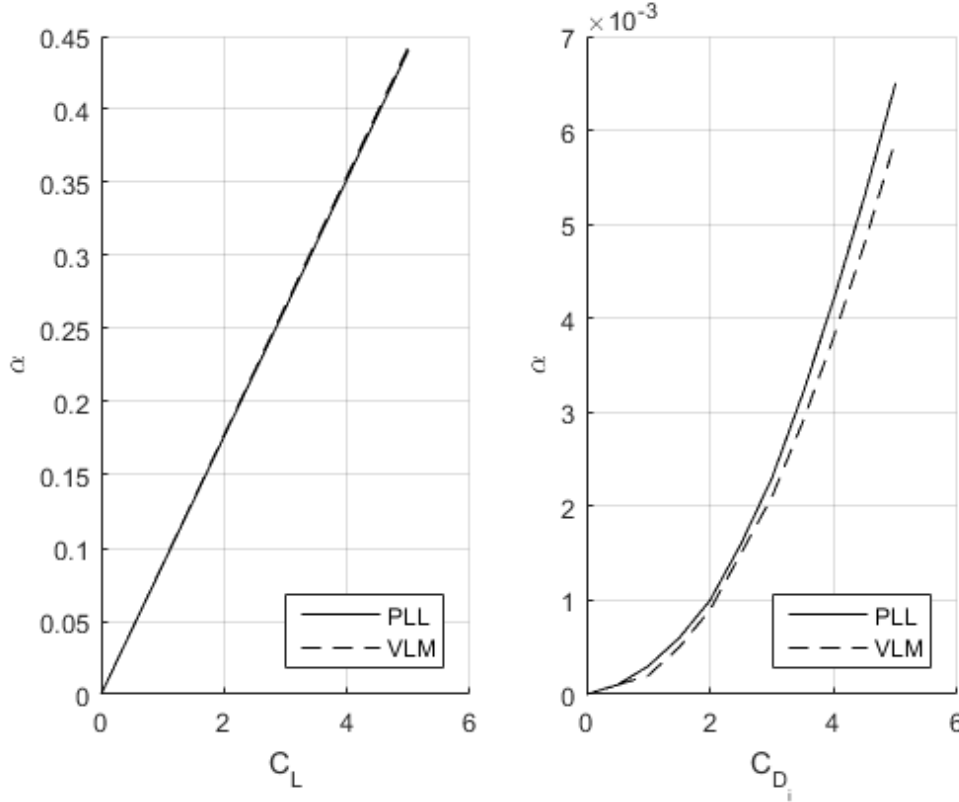
Şekil 2.12 : Kanadın paneller halinde gösterimi (birimler metre cinsindedir).

Analizlerden elde edilen sonuçlar ise aşağıda görüldüğü gibidir.

Çizelge 2.2 : Tek-kanat doğrulama testi için PLL ve VLM uygulamalarından elde edilen aerodinamik katsayılar.

α ($^\circ$)	PLL		VLM	
	C_L	C_{D_i}	C_L	C_{D_i}
0.0	0.0000	0.0000	0.0000	0.0000
0.5	0.0440	0.0001	0.0443	0.0001
1.0	0.0881	0.0003	0.0887	0.0002
1.5	0.1321	0.0006	0.1330	0.0005
2.0	0.1762	0.0010	0.1774	0.0009
2.5	0.2202	0.0016	0.2217	0.0015
3.0	0.2642	0.0023	0.2660	0.0021
3.5	0.3083	0.0032	0.3102	0.0029
4.0	0.3523	0.0042	0.3545	0.0038
4.5	0.3964	0.0053	0.3987	0.0048
5.0	0.4404	0.0065	0.4429	0.0059

Her iki yöntem için farklı hücum açılarında yürütülen analizden elde edilen taşıma ve indüklenmiş sürükleme katsayısı eğrilerinin nitel bir karşılaştırması da Şekil 2.13’de görüldüğü gibidir.

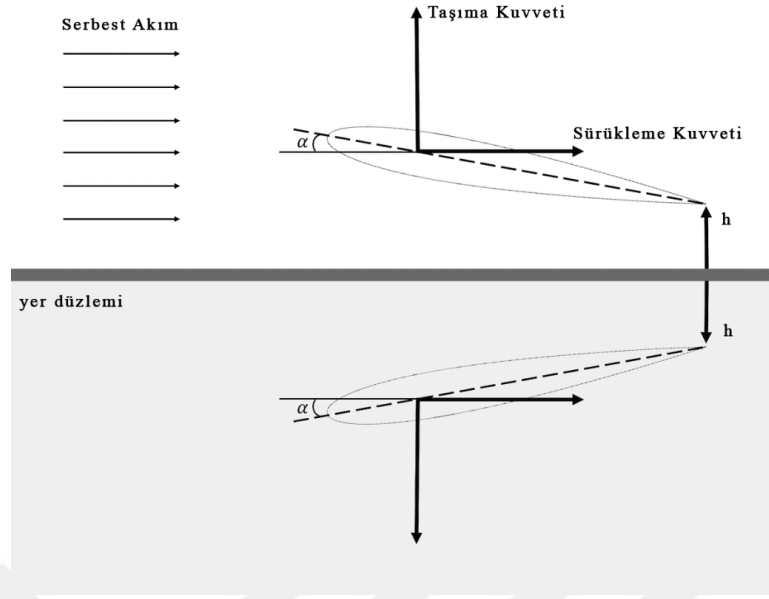


Şekil 2.13 : Doğrulama testi için PLL ve VLM sonuçlarının karşılaştırması.

Bu doğrulama testi sonucunda belirtilen VLM uygulamasının üç boyutlu düz bir kanat için PLL yöntemi ile gösterdiği uyumun, indüklenmiş sürükleme katsayısındaki hafif bir farklılığa rağmen, genel itibarıyla kabul edilebilir olduğu söylenebilir. PLL uygulamasında kullanılan segment sayısı ile VLM uygulamasında veter ve açıklık boyunca yerleştirilen panel sayısının bu farklılığa sebebiyet verdiği düşünülebilir.

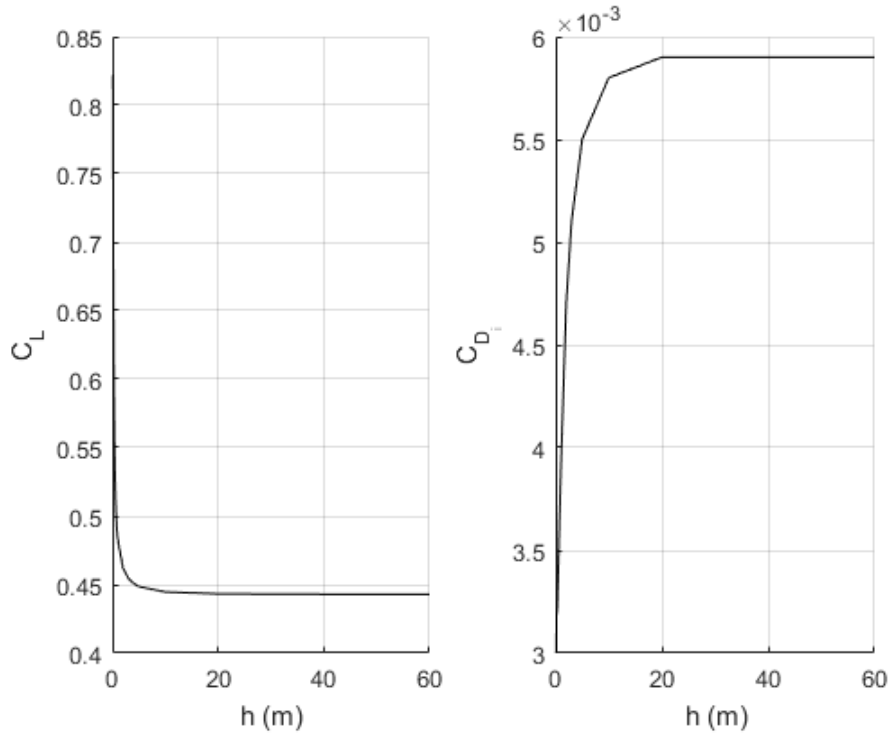
2.5.2 Yer etkisi analizi

Bilgisayar programının birden fazla taşıyıcı yüzey için testine yönelik olarak yapılan yer etkisi analizi uygulamasında, tek-kanat halindeki doğrulama testinde kullanılmış olan kanat Şekil 2.14 ‘de gösterildiği gibi simetrik ve ters açılı eşiyile birlikte aralarında belli bir uzaklık olacak biçimde yerleştirilmiştir. Yapılan uygulamada iki kanat arasındaki uzaklık arttırılarak taşıma katsayısındaki değişim gözlemlenmiştir.



Şekil 2.14 : Yer etkisi analizi arayüzü.

Hücum açısı 5 derece olmak üzere yapılan uygulamadan elde edilen sonuçlar Şekil 2.15’de sunulmuştur. Beklendiği gibi iki kanat arasındaki h uzaklığı küçük iken dikkati çekecek kadar büyük olan yer etkisi, bu uzaklık arttığında giderek azalmakta ve uzaklığın çok büyük değerlerinde kanatların bireysel taşıma ve indüklenmiş sürüklenme katsayıları tek-kanat halindeki değerlere asimptotik olarak yaklaşmaktadır. Bu sonuç husus programın iki kanat halinde doğru çalıştığını göstermektedir.



Şekil 2.15 : Yer etkisi analizi sonuçları.

2.5.3 Kanat-kuyruk sistemi analizi

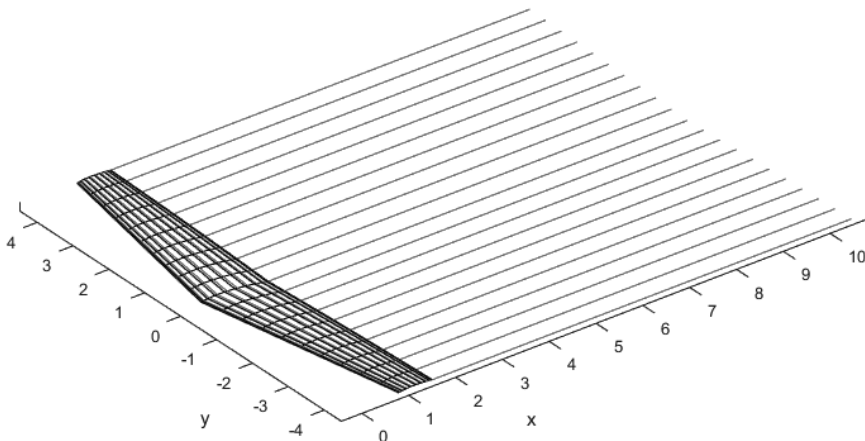
Bu uygulamada bir kanat-kuyruk sisteminin incelemesi ele alınmıştır. Öncelikle kambur profilli bir kanat tek başına analiz edilmiş ve sonra arkasına simetrik profilli ve sıfır oturma açılı bir kuyruk yerleştirilerek oluşturulan taşıyıcı yüzey sistemi incelenmiştir.

Kambur profilli yüzeyin oluşturacağı aşağı sapma nedeniyle, arkasındaki taşıyıcı yüzeyin etkin hücum açısının negatif yönde değişmesi ve bu sebeple de kuyruk için hesaplanacak olan taşıma katsayısının sıfırdan küçük olması beklenir.

Kanat kesiti olarak NACA 4412 kamburlu profili seçilmiştir. Bu kanat profili sıfır hücum açısında dahi pozitif bir taşıma katsayısına sahiptir. Şekil 2.16'deki gibi oluşturulan kanadın özellikleri Çizelge 2.3'deki gibidir. Serbest akım kanada sıfır derece açı ile gelmektedir.

Çizelge 2.3 : Kanat-kuyruk doğrulaması için oluşturulan kanadın özellikleri.

Girdi	Sembol	Değer
Kanat alanı	S_{ref}	$10 m^2$
Açıklık oranı	AR	8
Sivrilme oranı	λ	0.5
Geliş açısı	α_{in}	4°
Burulma açısı	α_{tw}	-4°
Ok açısı	α_{sw}	10°
Dihedral açısı	α_{di}	0°
Veter boyunca panel sayısı	n_i	10
Açıklık boyunca panel sayısı	n_j	20



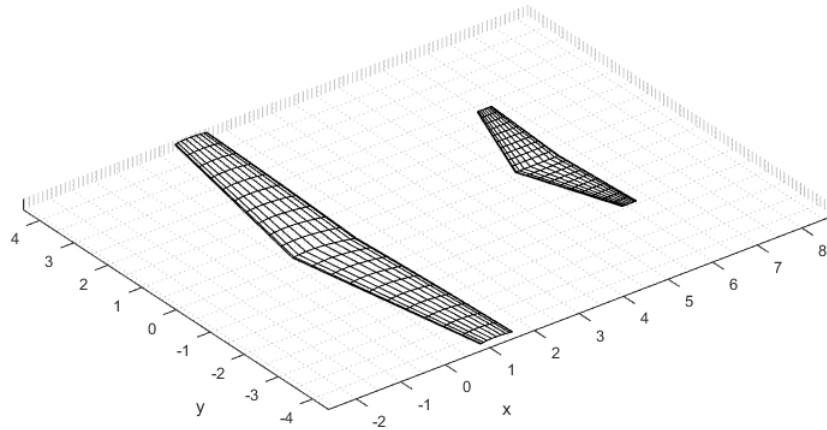
Şekil 2.16 : Kanat-kuyruk doğrulaması için oluşturulan kanat üzerindeki panel ve atnalı girdapları.

Çizelge 2.3’de belirtilen panel sayılarıyla oluşturulan kanat için VLM yöntemi uygulandığında elde edilen taşıma katsayısı 0.5016 olarak elde edilmektedir.

Daha önceden tanımlanan kanadın 5 metre gerisine yerleştirilen NACA 0012 simetrik profilli kuyruğun özellikleri Çizelge 2.4’de gösterildiği gibidir. Elde edilen kanat-kuyruk sistemi Şekil 2.17’de görüldüğü gibidir.

Çizelge 2.4 : Kanat-kuyruk doğrulaması için oluşturulan kuyruğun özellikleri.

Girdi	Sembol	Değer
Kanat alanı	S_{ref}	$3 m^2$
Açıklık oranı	AR	6
Sivrilme oranı	λ	0.33
Ok açısı	α_{sw}	20°
Geliş açısı	α_{in}	0°
Burulma açısı	α_{tw}	0°
Veter boyunca panel sayısı	n_i	10
Açıklık boyunca panel sayısı	n_j	20



Şekil 2.17 : Kanat-kuyruk doğrulaması için oluşturulan taşıyıcı yüzey sistemi üzerindeki paneller.

Sonuç olarak oluşturulan kanat-kuyruk sisteminin VLM ile analizinden elde edilen taşıma katsayıları Çizelge 2.5’deki gibi elde edilmiştir. Görüldüğü üzere, kanadın oluşturduğu sirkülasyonun etkisiyle yaratılan aşağı sapma ile kuyruğun maruz kaldığı efektif hücum açısının düşmesi ve buna bağlı olarak simetrik profilli ve sıfır oturma açısı ile yerleştirilen kuyruk için negatif taşıma katsayısı elde edildiği gözlemlenmiştir.

Çizelge 2.5 : Kanat-kuyruk sisteminin analizinden elde edilen sonuçlar.

	Kanat	Kuyruk	Toplam (kanat alanı referans alınmıştır)
C_L	0.4991	-0.1929	0.4412

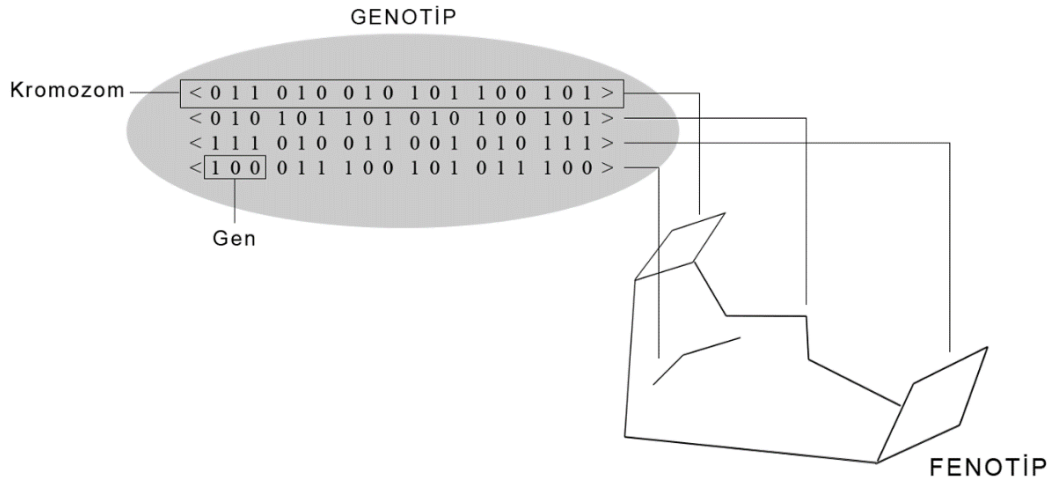


3. GENETİK ALGORİTMA

Bu başlık altında mühendislik problemleri için genetik algoritmaların genel uygulaması anlatılmaktadır. Taşıyıcı yüzey sistemi eniyilemesinde GA uygulanırken nelere dikkat edilmesi gerektiği ve gerçekleştirilmesi gereken adımların detayları işlenmiştir.

3.1 Taşıyıcı Yüzey Sistemi Eniyilemesi İçin Darvinci Evrim Teorisi

Literatür araştırması kısmında DET hakkında verilen tanımlamalar yapay sistemler için de geçerlidir. Yapay sistemi bir birey olarak ve kalıtsal (genetik) özelliklerini de birer gen olarak düşünelim. Bireyin sahip olduğu her gen topluluğuna kromozom adı verilir ve kromozomlar da bir bireyin genotipini oluşturur (Şekil 3.1). Genotipin dışa vurumuna da fenotip denir (Shiffman, 2012).



Şekil 3.1 : Yapay bir sistem için genotip ve fenotip gösterimi.

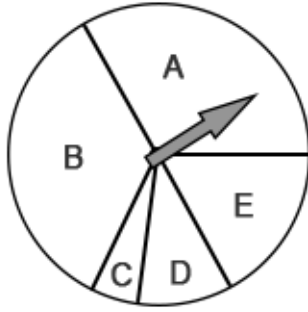
Rastgele oluşturulmuş veya elle tanımlanmış bireylerden oluşan bir popülasyon ele alınacak olursa, bu bireyler arasından bir sonraki neslin ebeveynlerinin seçilebilmesi için, bireylerin bir şekilde değerlendirilmesi ve puanlandırılması gerekir. Bu puanlanma sırasında bireylerin genotipleri, belli bir amaç doğrultusunda yazılan uygunluk fonksiyonları (fitness function) ile incelenir. Yüksek uygunluk değerine sahip bireyler, yeni neslin ebeveynleri olmaya hak kazanır.

Örneğin, Çizelge 3.1 ‘deki gibi 5 bireyden oluşan popülasyonu ele alalım. Hesaplanan uygunluk değerlerini birbirleri arasında karşılaştırabilmek adına, hepsini normalize etmek tavsiye edilir. Buradaki amaç sadece 0 ve 1 arasında uygunluk değerleri ile çalışmaktır. Bunun için her uygunluk değeri toplam ile bölünebilir. Normalize uygunluk değerleri 100 ile çarpılırsa, her bireyin ebeveyn olma olasılığı yüzde cinsinden bulunabilir.

Çizelge 3.1 : Örnek bir popülasyon için uygunluk değeri çizelgesi.

Birey	Uygunluk	Normalize Gösterim	Yüzde Gösterim
A	6	0.30	%30
B	8	0.40	%40
C	1	0.05	%5
D	2	0.10	%10
E	3	0.15	%15

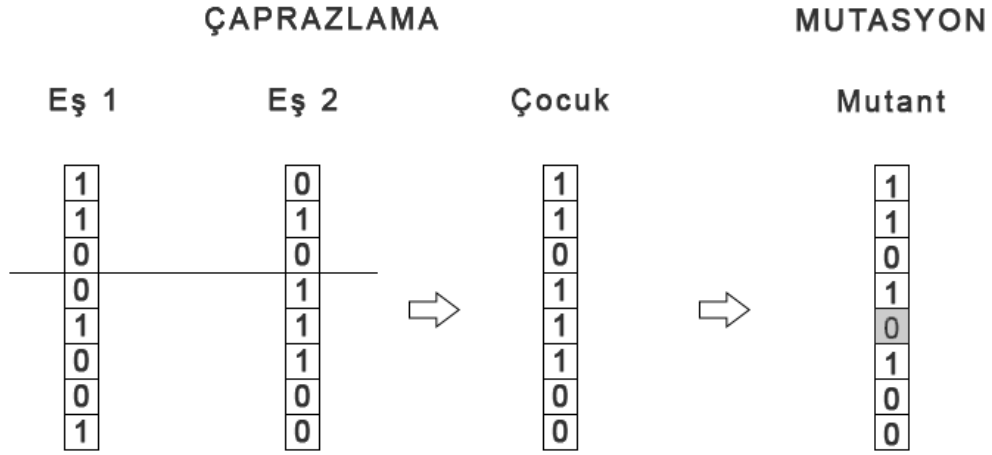
Bireyler arasından ebeveyn seçimi için bir eşleşme havuzu (mating pool) oluşturmalıdır. Bu havuzda yüksek uygunluk değerine sahip bireylerin eşleşme olasılığı uygunluk ile orantılı olmalıdır. Oluşturulan eşleşme havuzunda “çarkı çevir” yöntemi uygulanır ve eşleştirilecek bireyler olasılıksal olarak bulunur (Şekil 3.2).



Birey	Olasılık
A	%30
B	%40
C	%5
D	%10
E	%15

Şekil 3.2 : Çarkı çevir yöntemi.

Eşler seçildikten sonra, eş bireylerin genotiplerinin çaprazlama (crossover) ve mutasyon evrelerinden geçmesi ile yeni nesle ait bireyin genotip elde edilir. Çaprazlama ve mutasyon evreleri, GA uygulamalarında en çok değişkenlik gösteren evrelerdir. Doğada çaprazlama genel olarak kromozomların belli bir noktadan kesilmesi ve kesilen farklı parçaların birleştirilmesiyle etmesiyle sonuçlanır (Şekil 3.3). Mutasyon ise tamamen olasılıksaldır ve herhangi bir genin değiştirilmesine sebep olabilir. Yapay sistemlerde daha farklı uygulamalar mümkündür.



Şekil 3.3 : Çaprazlama ve mutasyon evreleri gösterimi.

Seçilimin ardından çaprazlama ve mutasyon evreleri tamamlandıktan sonra, elde edilen yeni nesil için Şekil 1.4'deki döngü devam ettirilir.

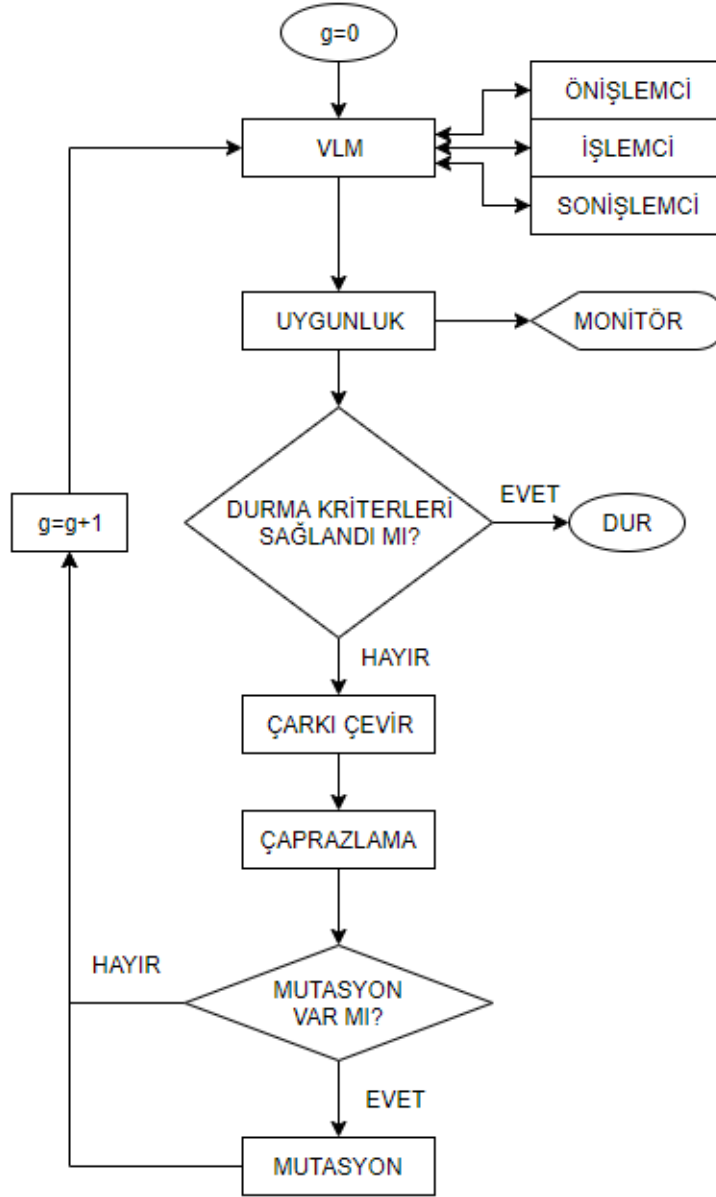
3.2 Programlama

Çok elemanlı yüzey sistemlerinin GA ile eniyilemesi için daha önce anlatılan aşamaların bir bilgisayar programına dönüştürülmesi bu başlık altında anlatılmaktadır. Şekil 1.4 esas alınarak, taşıyıcı yüzeylerin DET prensipleri çerçevesinde eniyilemesi için yazılacak bir bilgisayar programı için Şekil 3.4'deki akış çizelgesi oluşturulmuştur.

Bu akış çizelgesinde, bireylerin değerlendirilmesi aşaması VLM ile yüzeylerin aerodinamik analizi ve elde edilen katsayılar ile uygunluk hesabının yürütülmesi ile gerçekleştirilmiştir. Seçim ise sırasıyla Çarkı-Çevir, çaprazlama ve mutasyon evreleri vasıtasıyla sağlanmıştır.

Genetik algoritmali eniyileme programında, VLM programındaki gibi bir ana program ve alt-programlar düşünülmüştür. Ana programda evrimsel döngülerin yürütülmesi ve alt-programların çağırılması gibi işlemler yapılırken, diğer işlemlerin alt-programlar aracılığıyla yürütülmesi düşünülmüştür.

Evrimsel döngüler sırasında, ana programa girilen DET prensipleri çerçevesinde yazılan durma kriterlerine ek olarak hedef aerodinamik katsayı gibi farklı kriterler de yazılabilmektedir.



Şekil 3.4 : Taşıyıcı yüzeylerin eniyilemesi için akış çizelgesi.

3.2.1 Rastgele birey oluşturma

Popülasyondaki birey sayısı belirtildikten sonra bireylerin genotipinin rastgele oluşturulması gerekir. Genotiplerin alt parçaları olan kromozomların sayısı her bireyin sahip olduğu taşıyıcı yüzey kadardır ve kromozomlara ait genler yüzeylerle ilgili (2.22) denkleminde belirtilen bilgileri taşır.

Aşağıdaki gibi tanımlanan bir taşıyıcı yüzeyi düşünelim:

$$\text{yüzey} = \{4412, 10, 6, 0.6, 2, -2, 10, 0, 0, [0\ 0\ 0], 5, 10, 0\} \quad (3.1)$$

Bu sistem incelendiğinde 1 yüzeye sahip olduğu, yani tek kromozomlu olduğu anlaşılır. Kromozoma ait her bir eleman birer gen olarak ifade edilir. Bu tanımlı yapay sistemin genetik eniyilemesi öncesinde, hangi genlerde değişime izin verileceğinin yazılan bilgisayar programı içerisinde belirtilmesi gerekmektedir.

(3.1) denklemindeki kanada ait kromozomun 3. ve 7. genleri arası dâhil ve kalıtsal olarak tanımlanırsa, bu parametrelerin alabileceği değer aralıkları Çizelge 3.2’de gösterildiği gibi verilebilir. Oluşturulacak popülasyonun herhangi bir bireyinin genotipi girilen aralıklarla uyumlu olmalıdır.

Çizelge 3.2 : Örnek değişken gen parametreleri.

Parametre	Minimum Değer	Maximum Değer
Açıklık oranı	1	10
Sivrilme oranı	0	1
Oturma açısı	0°	5°
Burulma açısı	-5°	0°
Ok açısı	0°	10°

3.2.2 Bireylerin değerlendirilmesi

Bir popülasyondaki bireylerin değerlendirilmesi ve en güçlülerinin belirlenmesi sürecinde iki önemli aşama vardır. İlk olarak, Şekil 3.4’de görüldüğü gibi, popülasyondaki her bir bireyin genotipi VLM alt-programına gönderilir. VLM ile analiz edilen taşıyıcı yüzeylerden elde edilen aerodinamik katsayılar ana programa geri iletilir. Bireylere ait aerodinamik katsayılar uygunluk fonksiyonlarında kullanılır.

Uygunluk fonksiyonları GA uygulamalarının en önemli parçalarından biridir. Bu fonksiyonlar popülasyondaki güçlü bireyleri ayırt edilmesini sağlar. Yüksek uygunluk değerine sahip bireyler popülasyonun en güçlü bireyleridir ve genlerini sonraki nesillere aktarmaları sağlanmalıdır.

Bir önceki başlıkta verilen örneğe devam edilecek olursa, en uygun kanadın seçimi için aşağıdaki uygunluk fonksiyonu yazılabilir:

$$uygunluk = \frac{C_L}{C_{D_i}} \quad (3.2)$$

Bu uygulamada taşımanın indüklenmiş sürüklemeye oranı yüksek olan bireylerin uygunluk değeri daha yüksek olacaktır.

Uygunluk deęerleri hesaplandıktan sonra bütün deęerler 0 ile 1 arasında olacak şekilde normalize edilmelidir. Bunun için bütün uygunluk deęerlerinden minimum deęer çıkarılır ve maksimum uygunluk deęerine bölünür:

$$\text{normalize uygunluk}(s) = \frac{\text{uygunluk}(s) - \text{minimum uygunluk}}{\text{maksimum uygunluk}} \quad (3.3)$$

3.2.3 Durma kriterleri

Doęada, bir popülasyondaki erkek veya diři bireylerden biri eř bulabildięi sürece evrimsel döngü devam eder. Yapay sistemlerde erkek-diři ayrımı yapılmayabilir. Ancak, popülasyondaki kalıtsal bilgiler deęişim göstermiyorsa, nesillerin devam etmesinin bir anlamı olmayacaktır. Bunun sebebi DET prensiplerinden biri olan deęişimin sağlanması koşuludur.

Deęişimin hesaplanması ve kontrol edilmesi gerekmektedir. Bir popülasyondaki deęişim, neslin ortalama uygunluęunun bir önceki neslin ortalama uygunluęundan çıkarılıp oranlanması ile hesaplanabilir:

$$\text{deęişim}^g = \text{ortlama uygunluk}^g - \text{ortlama uygunluk}^{g-1} \quad (3.4)$$

(3.4) denkleminde deęişim, esasında her nesle karşılık gelen ortalama uygunluk eğrisinin eğimidir. Uygunluk eğrisinin deęişmemesi, o popülasyonda deęişimin gözlemlenmedięi anlamına gelir. Deęişim deęeri sıfır olduęunda bilgisayar programı durdurulur.

Deęişim kuralı dışında, bilgisayar programına nesil sayısı sınırı veya hedef uygunluk deęeri gibi ek durma kriterleri de eklenebilir.

3.2.4 Çaprazlama ve mutasyon

Uygunlukları hesaplanan bireylerin eşleřtirmesi sürecinde bir eşleme havuzu oluşturulur. Örneęin, 5 bireyden oluşan bir popülasyon düşünelim. Normalize uygunluk deęerleri toplam uygunluk deęerine bölünür ve 100 ile çarpılır. Bu şekilde bireylerin baskınlık deęerleri hesaplanmış olur. Baskınlık deęerleri her bireyin eş olabilme potansiyelini belirtir.

Çizelge 3.3 : Örnek bir popülasyona ait normalize uygunluk ve baskınlık değerleri.

Birey	Normalize Uygunluk	Baskınlık	Eşleşebilme Olasılığı
A	0.00	0	%0
B	0.58	29	%29
C	0.30	15	%15
D	1.00	50	%50
E	0.12	6	%6

Çizelge 3.3'deki gibi bir popülasyon için oluşturulan eşleşme havuzu dizinine 29 tane B bireyi, 15 tane C bireyi, 50 tane D bireyi ve 6 tane E bireyi eklenir. Havuz içerisinden *datasample* komutu ile rastgele seçilecek 2 farklı bireyler eşleştirilir.

Bu örnekte, baskınlık değerleri her bir bireyin eş olabilmeye oranının yüzdesi olarak gelmektedir. Ancak, farklı uygulamalar da mümkündür. Asıl amaç 0 ile 1 arasında olan normalize uygunluk değerleri kullanılarak bir havuz oluşturulması ve bu havuzdaki bireylerin eşleşme olasılığının uygunluk değerleri ile orantılı olmasıdır.

Eş seçiminden sonra, seçilen eşlerin genotiplerinin çaprazlanması aşaması gelir. Bu aşamada birçok uygulama mevcuttur. Ancak bu tez kapsamında değişimin tamamen rastgele bir şekilde yapılabilmesi için Şekil 3.3'deki gibi kromozomların belli bir yerden kesilip birleştirilmesi yerine farklı bir işlem uygulanmıştır. Her kromozomdaki değişken gen sayısı kadar dönecek bir döngüde, [1 2] arasından rastgele bir tam sayı MATLAB programlama dilinin *randi*([1 2]) komutu ile çağrılır. Eğer 1 değeri gelirse birinci eşin geni, 2 gelirse ikinci eşin geni sonraki nesle aktarılmak üzere saklanır.

Çaprazlama sonunda elde edilen genotipe çocuk genotip adı verilir. Ancak burada dikkat edilmesi gereken bir husus, *randi* komutu çağrılırken sürekli 1 veya 2 değerinin gelmediğinden emin olunmasıdır. Aksi takdirde, çocuk birey ebeveynlerinden birinin kopyası olur ve bu durumda değişim prensibi sağlanamış olur.

Son olarak, ana programda tanımlı mutasyon oranı gözetilerek çaprazlama sonucu elde edilen çocuk bireyler mutasyona uğratılır. Bu işlem için 0 ile 1 arasında rastgele bir gerçel sayı *rand* komutu ile çağrılır. Eğer çağrılan sayı mutasyon oranından küçük veya eşit ise, çocuk genler mutasyona uğratılmak üzere yazılan mutasyon alt-programına gönderilir. Bu program çocuk genotipine ait değişken genleri rastgele değiştirir ve ana programa geri iletir.

3.2.5 Gen aktarımı

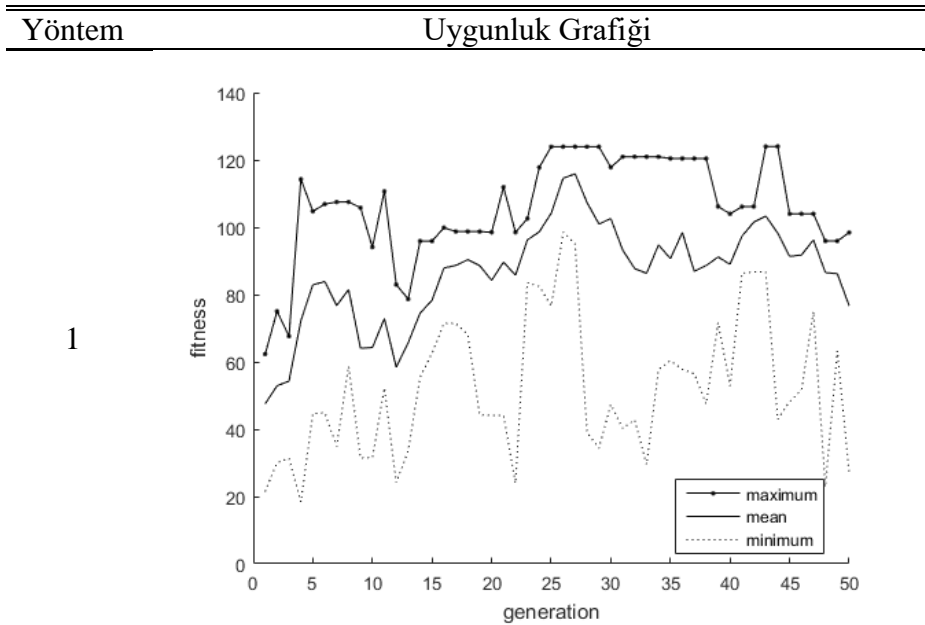
Çaprazlama ve mutasyondan sonra elde edilen çocuk genlerin sonraki nesle aktarımı için birçok yöntem uygulamak mümkündür. Doğada, her çocuk popülasyona ilave

birer bireydir ve yaşamlarını ölene kadar sürdürürler. Yapay sistemlerde bu işlem istenildiği gibi kontrol edilebilir. Bu başlık altında yeni doğan bireylerin popülasyona nasıl ekleneceği konusu işlenmektedir.

1. Veraset: İki bireyin eşleşmesinden elde edilen çocuk birey, bir sonraki nesilde birinci ebeveynin yerini alır.
2. Kontrollü veraset: Bu yöntemde verasetten farklı olarak, çocuk bir sonraki nesle eğer birinci ebeveyninden daha yüksek uygunluk değerine sahip ise eklenir.
3. Sağkalım: Adından da anlaşılacağı gibi, doğadaki gen aktarımı esas alınır. Çocuk bireyler, popülasyondaki en düşük uygunluk değerine sahip bireyin yerini alır. Eğer çocuğun uygunluk değeri en zayıf bireyinkinden de düşük ise, bu çocuk bir sonraki neslin popülasyonuna eklenmez.

Bu 3 farklı yöntemin uygulanması evrimsel döngülerin verimini ve hesaplamalara harcanan zamanı oldukça etkilemektedir. (3.1) denkleminde tanımlı kanat ve (3.2) denkleminde tanımlı uygunluk fonksiyonu kullanılarak elde edilen Çizelge 3.4'de her bir farklı yöntem için elde edilen maksimum, ortalama ve minimum uygunluk eğrilerindeki değişimler gösterilmiştir. Bu eniyileme döngülerinde popülasyon 10 bireyle sınırlı tutulmuş ve mutasyon oranı %20 girilmiştir. Maksimum nesil ise 50 girilmiştir.

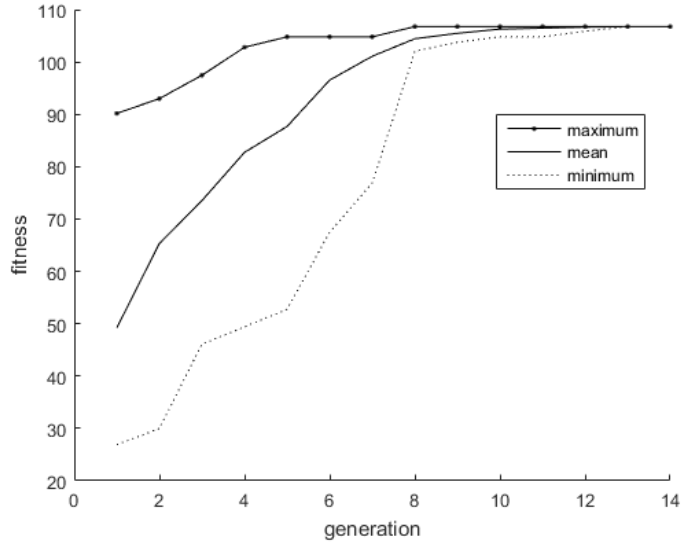
Çizelge 3.4 : Farklı gen aktarma yöntemlerinin karşılaştırması.



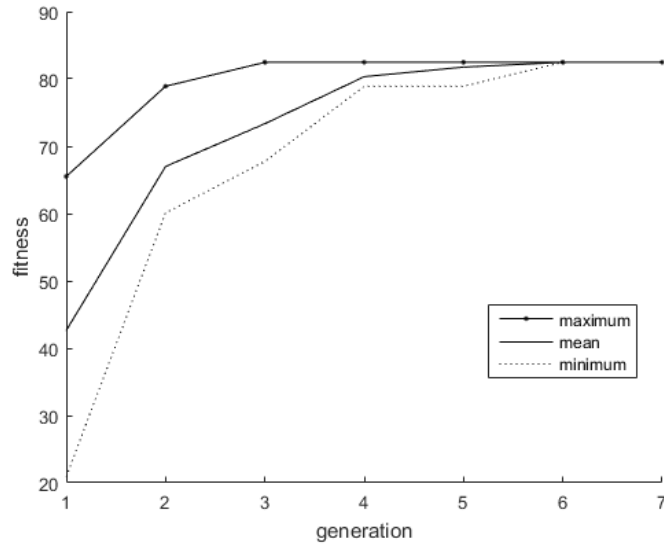
Çizelge 3.4 (devam) : Farklı gen aktarma yöntemlerinin karşılaştırması.

Yöntem	Uygunluk Grafiği
--------	------------------

2



3



Veraset yöntemiyle yürütülen eniyileme, Çizelge 3.4’te görüldüğü gibi 50. nesle kadar durmamıştır ve dolayısıyla harcanan süre bakımından diğer yöntemlere göre daha maliyetlidir. Ancak maksimum uygunluk bakımından diğer yöntemlere zaman zaman biraz daha yüksek değerlere ulaşmıştır. Kontrollü veraset 14. nesilden itibaren değişim prensibini sağlayamayarak durmuştur. Veraset yönteminden farklı olarak, sürekli artan uygunluk değerlerine sahiptir, bunun sebebi de zayıf ebeveynlerin yerini daha güçlü çocukların almasıdır. Son olarak, “sağkalım” yönteminin uygulandığı eniyilemede, kontrollü verasete benzer bir eğilim görülmektedir. Daha az nesilde durma kriterlerini sağlayan bu yöntem zamandan kazandırsa da kontrollü veraset yönteminden daha düşük uygunluk değerleri vermiştir.



4. UYGULAMALAR

Çok elemanlı taşıyıcı yüzey sistemlerinin VLM ile incelemesi ve GA vasıtasıyla eniyilemesi, önceki başlıklar altında genel olarak incelenmiştir. Bu kısımda ise birbirini takip eden iki örnek ile birlikte bu eniyileme yönteminin uygulaması detaylı bir şekilde incelenmiştir. Öncelikle tek yüzeyden oluşan bir kanadın eniyilemesinin ardından, arkasına yerleştirilen bir diğer yüzey ile oluşan kanat-kuyruk sisteminin eniyilemesi gerçekleştirilmiştir.

4.1 Kanat Eniyilemesi

Deniz seviyesinde ve düşük hızlarda uçuşu için tasarlanmış bir kanat düşünelim. NACA 4412 kesit profilli bu kanat, seyir uçuşu için $C_{LC} = 0.40$ taşıma katsayısına ve aşağıdaki özelliklere sahip olsun.

Çizelge 4.1 : Kanat eniyilemesi için verilen orijinal parametreler.

Parametre	Değer
Profil	4412
Kanat alanı	10
Açıklık oranı	7
Sivrilme oranı	1
Oturma açısı	2°
Burulma açısı	-1°
Ok açısı	0°
Dihedral açısı	0°

Bu kanat üzerinde GA kullanarak eniyileme yapmak istersek, tanımlanan $C_{LC} = 0.40$ değerini sağlayacak bir uygunluk fonksiyonunun yazılması gerekir. Oluşturulacak rastgele popülasyonda bu değer altında taşıma katsayısına sahip bireyler popülasyondan çıkartılırken, yakın değere sahip bireylerin uygunluk değeri yüksek verilmelidir.

$y = 1/x$ fonksiyonu, artan x değerleri için azalan y sonucu verir. Taşıma katsayısı için böyle bir uygunluk fonksiyonu kullanılabilir. Ancak, sürüklemeyi düşürmek de kanat eniyilemesinde dikkat edilen bir husustur. Yüksek indüklenmiş sürüklenme

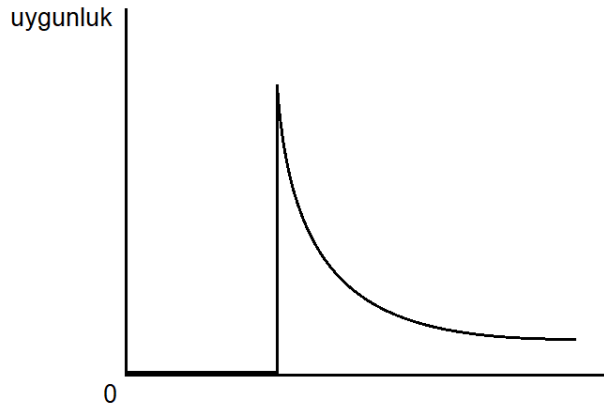
katsayısına sahip bireyleri seçim sürecinde zayıflatmak için, taşıma katsayısı için verilen fonksiyon indüklenmiş sürüklemeye de bölünür ise, sonuç olarak elde edilen uygunluk fonksiyonu güçlü bireylerin tanımlanmasında kullanılabilir:

$$uygunluk = \frac{1}{C_L C_{D_i}} \quad (4.1)$$

$C_{LC} = 0.40$ değerinden küçük taşıma katsayılı bireyleri engellemek için ise uygunluk fonksiyonuna kod içerisinde bir *if* ifadesi eklemek yeterli olacaktır:

```
fitness = 1/CL/CD_i;
if CL < 0.40
    fitness = 0;
end
```

Bu şekilde elde edilecek olan uygunluk fonksiyonunun grafiği değişen taşıma ve indüklenmiş sürükleme katsayıları için aşağıdaki gibi olacaktır.



Şekil 4.1 : Kanat eniyilemesi için uygunluk fonksiyonu.

Uygunluk fonksiyonu tanımlandıktan sonra, oluşturulacak popülasyonun boyutu ve mutasyon oranı tanımlanır. Bu uygulama için popülasyon 10 bireyle sınırlı tutulmuş olup mutasyon oranı %20 girilmiştir. So olarak maksimum nesil 100 girilmiştir.

Eniyileme yapılacak bireye ait genotip aşağıdaki gibi tanımlanmıştır:

$$kanat = \{4412, 10, 7, 1, 2, -1, 0, 0, 0, [0\ 0\ 0], 10, 10, 0\} \quad (4.2)$$

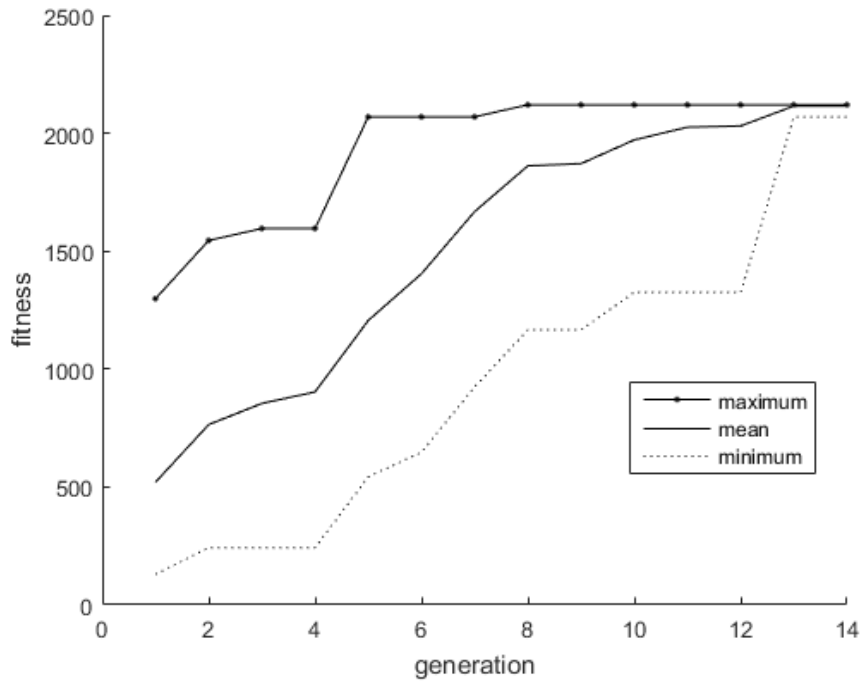
VLM çözümlerinde kanat için veter doğrultusunda 10 panel, açıklık boyunca ise 10 panel düşünülmüştür ve tek kanat üzerinde çalışıldığı için kanadın yarısı üzerinde analizlerin yürütülmesi yeterlidir.

(4.2) denkleminde genotipi verilen orijinal kanadın açıklık oranı, sivrilme oranı, oturma açısı ve ok açısı parametrelerini tanımlayan genler kalıtsal olarak tanımlanmıştır. Bu genler için ise Çizelge 4.2'deki değişim aralıkları verilmiştir.

Çizelge 4.2 : Kanat eniyilemesi için değişim aralıkları.

Parametre	Minimum Değer	Maksimum Değer
Açıklık oranı	5	10
Sivrilme oranı	0	1
Oturma açısı	0°	5°
Burulma açısı	-5°	0°
Ok açısı	0°	10°

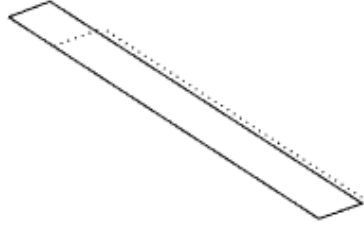
Kontrollü veraset gen aktarım yöntemi ile yürütülen nesiller sırasında popülasyondaki uygunluk değerlerindeki değişim ekrana grafik olarak yansıtılarak incelenmiştir.



Şekil 4.2 : Kanat eniyilemesi uygunluk grafiği.

Şekil 4.2'den de görüldüğü gibi, eniyileme yedinci nesilde değişim prensibini sağlayamayarak durmuştur. Sonuçta en güçlü olarak çıkan 1 numaralı birey de ekrana yansıtılmış ve kanat geometrisindeki değişim gözlemlenmiştir (Şekil 4.3).

Orijinal kanat ve eniyileme sonucu elde edilen kanada ait parametrelerin karşılaştırması da Çizelge 4.3'de görüldüğü gibidir. Buradan görüldüğü gibi, GA kalıtsal genlerde değişim yaparken diğerlerini sabit tutmuştur.



Şekil 4.3 : Kanat eniyilemesinden elde edilen son geometri (kesikli çizgi orijinal kanattır).

Çizelge 4.3 : Orijinal ve en iyi kanat karşılaştırması.

	Parametre	Orijinal Kanat	Eniyilenmiş Kanat
GENOTİP	<i>profil</i>	4412	4412
	<i>S</i>	10	10
	<i>AR</i>	7	9.55
	λ	1	0.95
	α_{in}	2°	1.59°
	α_{tw}	-1°	-1.53°
	α_{sw}	0°	0°
	α_{di}	0°	0°
	<i>H</i>	0	0
	<i>R</i>	$[0\ 0\ 0]^T$	$[0\ 0\ 0]^T$
	n_i	10	10
	n_j	10	10
	<i>SS</i>	0	0
	VLM	C_L	0.4372
C_{D_i}		0.0082	0.0054
GA	<i>uygunluk</i>	1298	2120

4.2 Kanat-Kuyruk Sistemi Eniyilemesi

Bu uygulamada, bir önceki uygulamada eniyilenen kanadın arkasına bir yatay kuyruğun eklenmesini inceleyelim. Eklenen rastgele bir yatay kuyruğun GA ile eniyilemesini ele alalım.

Yine ilk olarak uygunluk fonksiyonu tanımlanmalıdır. Yatay kuyruklar uçuş esnasında uçağın uzunlamasına kararlılığının sağlanmasına yarar. Bunu, uçağın yunuslama momentini sıfırlayarak yaparlar. Geliştirilen VLM kodlamasında hesaplanan momenti sıfır yapacak olan kuyruklara sahip bireylerin uygunlukları artırılmalıdır veya sıfırdan uzaklaşan yunuslama momentleri için uygunluk azaltılmalıdır.

Bunun için (4.1) denkleminde belirtilen uygunluk fonksiyonuna ek olarak moment katsayısı da aşağıdaki gibi eklenebilir:

$$uygunluk = \frac{1}{C_L C_{D_i} C_{M_y}} \quad (4.3)$$

Uygunluk fonksiyonu tanımlandıktan sonra, kuyruğun nereye yerleştirileceği düşünülmelidir. Bu konu aslında tam bir sistem mühendisliği problemidir ve bu yerleşim parametresi için de uygunluk fonksiyonu düzenlenebilir. Hatta, kuyruğun yaratacağı yunuslamanın sistemin ağırlık merkezi etrafında hesaplanması gerekmektedir, ancak, bu uygulamada kolaylık sağlama açısından ağırlık merkezinin orijin noktasında olduğu varsayımı yapılmıştır. Ancak bu uygulamada şimdilik daha basit bir yaklaşım yaparak kuyruğun yerleşimi kanadın 5 metre gerisinde sabit tutulmuştur. Ek olarak belirlenen kuyruğa ait genotip aşağıdaki gibi girilmiştir:

$$kuyruk = \{0012, 3, 6, 0.5, 0, 0, 10, 0, 0, [5 \ 0 \ 0], 10, 10, 1\} \quad (4.4)$$

Çizelge 4.3'deki eniyilenmiş kanat genotipine ek bir kromozom olarak (4.4) denkleminde tanımlı kuyruk eklenmiş ve aşağıdaki kanat-kuyruk sistemine ait çift kromozomlu taşıyıcı yüzey sistemi tanımlanmıştır:

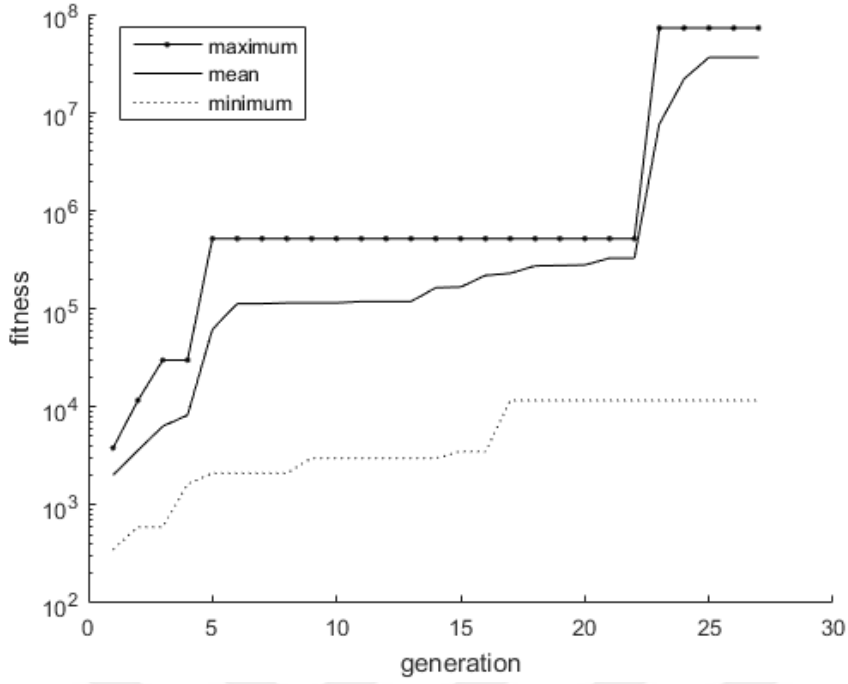
$$sistem = \left\{ \begin{array}{l} \{4412, 10, 9.55, 0.95, 1.59, -1.53, 0, 0, 0, [0 \ 0 \ 0], 10, 10, 1\} \\ \{0012, 3, 6, 0.5, 0, 0, 10, 0, 0, [5 \ 0 \ 0], 10, 10, 1\} \end{array} \right\} \quad (4.5)$$

Bu sistemde, eniyilenen kanat genleri yine Çizelge 4.2'deki aralıklarda değişken tutulup yatay kuyruk kromozomu üzerindeki genler için Çizelge 4.4'deki değişim aralıkları tanımlanmıştır.

Çizelge 4.4 : Kanat-kuyruk eniyilemesinde kuyruk için değişim aralıkları.

Parametre	Minimum Değer	Maksimum Değer
Açıklık oranı	1	8
Sivrilme oranı	0	1
Oturma açısı	-5°	0°
Burulma açısı	0°	5°
Ok açısı	0°	20°

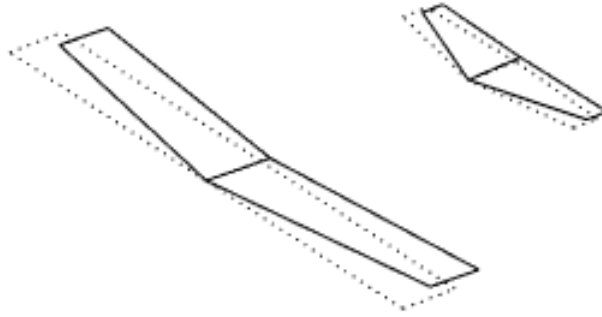
Yine kontrollü veraset yöntemiyle yürütülen eniyilemede uygunluk grafikleri Şekil 4.4'deki gibi izlenilmiştir. Nesillerin değişim prensibini sağlayamama durma kriteri sebebiyle 27. neslin sonunda durduğu gözlemlenmiştir.



Şekil 4.4 : Kanat-kuyruk eniyilemesi uygunluk grafiği.

Sonuç olarak, en iyi taşıyıcı yüzey sistemi popülasyonun 2 numaralı bireyi seçilmiş olup aşağıdaki gibi şekil almıştır.

2



Şekil 4.5 : Kanat-kuyruk eniyilemesinden elde edilen son geometri (kesikli çizgi orijinal kanat-kuyruk sistemidir).

Eniyilenmiş kanat-kuyruk sisteminin orijinal sistem ile karşılaştırması Çizelge 4.5'deki gibi verilmiştir.

Çizelge 4.5 : Orijinal ve eniyilenmiş kanat-kuyruk sistemi karşılaştırması.

	Parametre	Orijinal Kanat	Orijinal Kuyruk	Eniyilenmiş Kanat	Eniyilenmiş Kuyruk
GENOTİP	<i>profil</i>	4412	0012	4412	0012
	<i>S</i>	10	3	10	3
	<i>AR</i>	9.55	6	8.49	5.55
	λ	0.95	0.5	0.76	0.39
	α_{in}	1.59°	0°	1.98°	-2.65°
	α_{tw}	-1.53°	0°	-0.05°	2.19°
	α_{sw}	0°	10°	9.10°	18.71°
	α_{di}	0°	0°	0°	0°
	<i>H</i>	0	0	0	0
	<i>R</i>	[0 0 0] ^T	[5 0 0] ^T	[0 0 0] ^T	[5 0 0] ^T
	<i>n_i</i>	10	10	10	10
	<i>n_j</i>	10	10	10	10
	<i>ss</i>	1	1	1	1
	VLM	<i>C_L</i>	0.3737		0.4168
<i>C_{D_i}</i>		0.0043		0.0074	
<i>C_{M_y}</i>		-0.0120		0.0000	
GA	<i>uygunluk</i>	0		7.2 × 10 ⁷	



5. SONUÇ VE ÖNERİLER

Günümüzde yaygın kullanıma sahip hesaplamalı aerodinamik yöntemlerinden biri olan Girdap Kafes Yöntemi (VLM) ile taşıyıcı yüzeylerin incelemesi ve doğadaki “en güçlünün hayatta kalması” prensibi esas alınarak geliştirilen bir arama yöntemi olan Genetik Algoritmalarla (GA) eniyileme konuları bu çalışmada detaylı bir şekilde ele alınmıştır. Çok elemanlı taşıyıcı sistemlerin incelemesinde kullanılabilen VLM yöntemi ile kompleks yapay sistemlerin çok yönlü eniyilemesine olanak sağlayan GAların bir sentezi ile, matematiksel eniyileme yöntemlerine nazaran basit bir çalışma prensibine sahip, doğadan ilham alan, kolay programlanabilir ve oldukça dayanıklı bir bilgisayar programı geliştirilmiştir. MATLAB programlama diliyle geliştirilen bu bilgisayar programı öncelikle doğrulama testlerine tabi tutulmuş ve kullanılan yöntemlerin farklı mühendislik problemlerinin çözümündeki kabiliyeti uygulamalı bir şekilde incelenmiştir.

İlk olarak çok elemanlı taşıyıcı yüzey sistemleri için bir VLM formülasyonu yazılmıştır. Bu formülasyonu kullanan bir bilgisayar programı vasıtasıyla yürütülen düz bir kanat analizi, Prandtl’ın taşıyıcı Çizgi Yöntemi (PLL) ile karşılaştırılmış ve uyumlu sonuçlar elde edilmiştir. Bunun ardından, ters açılı yerleştirilmiş bir çift kanat sistemi vasıtasıyla yer etkisi ve bir kanadın arkasındaki kuyruk üzerinde aşağı sapmanın etkisiyle gerçekleşen etkin hücum açısındaki düşüş incelenmiştir.

Yapay sistemlerin eniyilemesinde GA kullanımı ile ilgili genel konular işlendikten sonra, Darvinci Evrim Teorisindeki (DET) üç temel prensip (kalıtım, değişim ve seçim) esas alınarak geliştirilen bir eniyileme uygulaması, daha önceden yazılan VLM programı da kullanılarak geliştirilmiştir. Rastgele bireylerden oluşan bir taşıyıcı yüzey sistemleri popülasyonunda, güçlü bireylerin belirlenmesinde uygulamanın amacına göre değişen uygunluk fonksiyonları yazılmıştır. İki bireyin eşleşmesi sonucu gerçekleşen çaprazmalama ve mutasyon evrelerini takiben, elde edilen çocuk genlerin bir sonraki nesle aktarımı için 3 farklı yöntem sunulmuştur: çocukların popülasyonda ebeveynlerinin koşulsuz yerini aldığı “veraset” yöntemi, bu yöntemin kontrollü bir şekilde gerçekleştiği ikinci bir yöntem ve güçlü çocukların zayıf bireylerin yerini

aldığı “sağkalım” yöntemi. Bu yöntemler arasından ikinci ve üçüncü yöntemler daha kontrollü bir şekilde gerçekleştikleri için en güçlü bireye daha hızlı yakınsamıştır. Ancak bu yakınsamanın gerçekten en iyi sonucu getirip getirmediği iyi incelenmelidir.

Son olarak, geliştirilen GAlı bilgisayar programı, tek kanat ve bir kanat-kuyruk sisteminin eniyilemesi problemleri için kullanılmıştır. İlk uygulamada eniyilenen kanadın arkasına, ikinci uygulamada bir kuyruk yerleştirilmiştir. Oluşturulan taşıyıcı yüzey sistemin eniyilemesinde kanadın bir önceki uygulamada eniyilenmesine rağmen değişime uğradığı gözlemlenmiştir. Bu bakımdan taşıyıcı yüzey sistemlerinin eniyilemesinde, sisteme ait elemanların ayrı ayrı eniyilemesinden ziyade, toplu halde ve birbirleri arasındaki aerodinamik etkileşimlerin hesaba katılmasının daha uygun olduğu anlaşılmıştır.

Herhangi bir yapay sistemin hızlı ve verimli bir şekilde eniyilemesine olanak sağlayan GAların dayanıklılığı da bu tez kapsamında verilen örneklerde doğrulanmış olup, uçak ve uzay mühendisliği problemlerine uygulanabilirliği konusunda ipuçları vermektedir. Geliştirilen eniyileme yönteminin kısıtlamaları ise GAnın kendisinden gelmeyip, çözüm yönteminin altında yatan teorik arkaplan olmuştur.

Sıkıştırılabilir etkiler ve vizkozite ile ilgili terimlerden yoksun potansiyel akışa bir çözüm sunan VLM yöntemi eniyilemenin uygulama alanını daraltmaktadır. Bu sebeple eniyilemeyi daha da geliştirmek için, standart VLM yönteminde hesaplanan indüklenmiş sürüklemeye vizkoz sürüklenme terimi eklenebilir. Kesit profili bilinen taşıyıcı yüzeye ait vizkoz sürüklenme verilerinin, VLM içerisinde hesaplanan etkin hücum açıları kullanılarak çağrılması yeterli olacaktır. Ayrıca yüksek hücum açılarındaki lineer olmayan taşıma bölgesinde de çalışabilmek adına, bu kapsamda geliştirilmiş VLM çözümleri de kullanılabilir. Aerodinamik performans parametrelerine ek olarak, esasında çok-disiplinli bir süreç olan tasarım için, yapısal ve kontrol ile ilgili performans parametrelerinin evrimsel döngülere uygunluk fonksiyonları kapsamında dâhil edilmesi, eniyilemenin kapsamını ve sonuç olarak elde edilen güçlü bireylerin güvenilirliğini artıracaktır.

Bir diğer kısıtlama ise bilgisayar gücüdür. Hava araçlarının konsept ve ön tasarım süreci kapsamında kullanılan VLM ve benzeri hesaplamalı çözüm yöntemleri, bilgisayar gücü açısından günümüz itibarıyla maliyetli sayılmamaktadır. Ancak GA uygulamasında birçok bireyden oluşan popülasyona ait taşıyıcı yüzeyler her nesilde

birer birer analiz edilmelidir. Bu sebeple de bilgisayar gücü ve çözüm için harcanan süre kayda değer bir hâl almaktadır. Örneğin, bu tez kapsamında yürütülen VLM doğrulama çalışmaları sırasında harcanan süre saniyeler mertebesindeyken, eniyileme uygulamaları dakikalar, hatta saatler mertebesinde gerçekleşebilmektedir. Ancak, her geçen gün daha da gelişen bilgisayar bilimi bu sorunu yakın gelecekte önemsiz kılacaktır. Günümüzde sıkça adı geçen kuantum bilgisayarların da geliştirilmesiyle katbekat artacak bilgisayar gücünün GA ve benzeri evrimsel programlama yöntemlerinin mühendislik uygulamalarında kullanımını yaygınlaştırması beklenebilir.





KAYNAKLAR

- Anderson, J. D.** (2016). *Fundamentals of Aerodynamics*. McGraw-Hill Education.
- Chang, S. S.** (2018). *A gentle introduction to genetic algorithms*. Retrieved May 2, 2018, from <https://sausheong.github.io/posts/a-gentle-introduction-to-genetic-algorithms/>
- Chong, E. K. P. & Zak, S. H.** (2013). *An introduction to Optimization*. John Wiley & Sons.
- DeYoung, J.** (1976). Vortex-lattice utilisation (Rapor No. 405). Virginia: NASA Raporu.
- Goldberg, D. E.** (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley Publishing Company, Inc.
- Jacobs, E. N., Ward, K. E. ve Pinkerton, R. M.** (1933). The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel (Rapor No. 460). Washington: NACA Raporu.
- Katz, J. & Plotkin, A.** (2001). *Low-Speed Aerodynamics*. Cambridge University Press.
- Ladson, C. L., Brooks, C. W. ve Hill, A. S.** (1996). Computer Program to Obtain Ordinates for NACA Airfoils (Rapor No. 4741). Virginia: NASA Raporu.
- Sadraey, M. H.** (2012). *Aircraft Design: A Systems Engineering Approach*. John Wiley & Sons.
- Shiffman, D.** (2012). The Nature of Code: Simulating Natural Systems with Processing. Retrieved from <http://natureofcode.com/>
- Shiffman, D.** (2017). 9.3: *Genetic Algorithm: Shakespeare Monkey Example - The Nature of Code*. Retrieved May 2, 2018, from <https://www.youtube.com/watch?v=nrKjSeoc7fc>
- Sivanandam, S. N. & Deepa, S. N.** (2007). *Introduction to Genetic Algorithms*. Springer-Verlag Berlin Heidelberg.
- Yükselen, M. A.** (2012). *Hesaplamalı aerodinamik ders notları*. Retrieved from http://web.itu.edu.tr/yukselen/UCK419/UCK419_Index.html



EKLER

EK A: GEN.m

```
%% GENETIC OPTIMIZATION CODE
% Author      : Osman Mirza Demircan
% Institution  : Istanbul Technical University
% Contact     : demircano16@itu.edu.tr
close all; clear; clc; format compact; tic
%% INPUT
disp('Loading input...')
% Freestream Parameters
global U_inf alpha
h      = 0.00e4; % altitude (m)
U_inf = 001.00; % airspeed (m/s)
alpha = 00.00; % angle (deg)
% Surface Definition
global specimen ref tail
specimen = {... % genotype
% profile S      AR      TR      a_in      a_tw      a_sw      a_di      H      R
ni nj ss
% {'4412'      010.00 07.00 1.00 +02.00 -01.00 +00.00 -00.00 0.00
[+00.00;+00.00;+00.00] 10 10 0}
{'4412'      010.00 09.55 0.95 +01.59 -01.53 +00.00 -00.00 0.00
[+00.00;+00.00;+00.00] 10 10 1}
{'0012'      003.00 06.00 0.50 +00.00 -00.00 +10.00 -00.00 0.00
[+05.00;+00.00;+00.00] 10 10 1}
};
ref = 1;
tail = 2;
% Optimization Parameters
global np mr gg sequence constraint mod
mg = 010; % minimum generation
ng = 100; % maximum generation
np = 010; % population size
mr = 0.20; % mutation rate
gg = 0.40; % goal lift coefficient
sequence = {... % genetic sequence
(3:7)
(3:7)
};
constraint = {... % genetic constraint
{[05.00 10.00] [0.00 1.00] [-00.00 +05.00] [-05.00 +00.00] [00.00
+10.00]}
{[01.00 08.00] [0.00 1.00] [-05.00 +00.00] [-00.00 +05.00] [00.00
+20.00]}
};
mod = 2; % 1: heritance || 2: controlled heritance || 3: survival of
the fittest
% Interface Parameters
global flag
flag = 0; % 0 for fittest | 1 for all
if flag == 1
    n_row = 5; % number of rows
    n_col = 8; % number of columns
end
disp('Loading completed.')
%% PREPROCESSING
% Freestream Parameters
global rho_inf q_inf V_inf
```

```

[~,~,~,rho_inf] = atmosisa(h);
q_inf = 0.5*rho_inf*U_inf^2;
V_inf = [U_inf*cosd(alpha);0;U_inf*sind(alpha)];
% Surface Parameters
global ns
ns = size(specimen,1);
disp('Evaluating initial specimen...')
[CL_0,CD_i_0,CM_0] = VLM(specimen);
fitness_0 = GEN_FITNESS(CL_0,CD_i_0,CM_0);
disp('Evaluation completed.')
%% GENETIC OPTIMIZATION
disp('Generating population...')
POP = cell(np,1);
POP{1} = specimen;
FIT = zeros(np,1);
FIT(1) = fitness_0;
F1 = figure(1);
set(F1,'name','Fitness Monitor','numbertitle','off')
F2 = figure(2);
set(F2,'name','Phenotype Monitor','numbertitle','off')
FIT_max = [];
FIT_mean = [];
FIT_min = [];
var1 = 1;
var2 = 1;
g = 0;
while g < ng
    g = g+1;
    disp(['Generation: ',sprintf('%3.0f',g)])
    % Population Check
    for p = 1:np
        while FIT(p) == 0
            POP{p} = GEN_NEWBORN;
            [CL,CD_i,CM] = VLM(POP{p});
            FIT(p) = GEN_FITNESS(CL,CD_i,CM);
        end
    end
    % Fitness Monitoring
    FIT_max = [FIT_max;max(FIT)];
    FIT_mean = [FIT_mean;mean(FIT)];
    FIT_min = [FIT_min;min(FIT)];
    if g > mg
        var1 = FIT_mean(end)-FIT_mean(end-1);
        var2 = FIT_mean(end-1)-FIT_mean(end-2);
    end
    figure(1)
    hold on
    plot(1:g,FIT_max,'k.-')
    plot(1:g,FIT_mean,'k-')
    plot(1:g,FIT_min,'k:')
    xlabel('generation')
    ylabel('fitness')
    legend('maximum','mean','minimum','Location','Best')
    if FIT_max(end)/FIT_max(1) > 1e6
        set(gca,'YScale','log')
    end
    figure(2)
    clf(2)
    if flag == 0
        hold on
        GEN_PHENOTYPE(specimen,':k')
    end
end

```



```

        id_fit = find(FIT==max(FIT));
        GEN_PHENOTYPE(POP{id_fit(1)}, '-k')
        title(sprintf('%3.0f', id_fit(1)))
        axis equal off
        view(3)
    else
        for n = 1:np
            hold on
            subplot(n_row, n_col, n)
            GEN_PHENOTYPE(POP{n}, '-k')
            title(sprintf('%3.0f', n))
            axis equal off
            view(3)
        end
        hold off
    end
    drawnow
    % Stopping Criteria
    if abs(var1) <= 0.01 && abs(var2) <= 0.01
        disp('Variation principle is compromised.')
        break
    elseif g == ng
        disp('Generation limit is reached.')
        break
    else % Natural Selection
        [POP, FIT] = GEN_SELECTION(POP, FIT);
    end
end
disp('Generation completed.')
id_1 = find(FIT==max(FIT));
fitness_1 = FIT(id_1(1));
optimization = (fitness_1 - fitness_0) / fitness_0;
optimum = POP{id_1(1)};
[CL_1, CD_i_1, CM_1] = VLM(optimum);
%% OUTPUT
disp('Outputting results...')
disp('Optimization Summary:')
disp(['Population size is ', sprintf('%0.0f', np)])
disp(['Mutation rate is ', sprintf('%0.2f', mr)])
disp(['Initial fitness is ', sprintf('%0.2f', fitness_0)])
disp(['Final fitness is ', sprintf('%0.2f', fitness_1)])
disp(['Optimization rate is
', sprintf('%0.0f', optimization*100), '%'])
if flag == 1
    F3 = figure(3);
    set(F3, 'name', 'Fittest Specimen', 'numbertitle', 'off')
    figure(3)
    hold on
    GEN_PHENOTYPE(specimen, ':k')
    GEN_PHENOTYPE(POP{id_1(1)}, '-k')
    title(sprintf('%3.0f', id_1(1)))
    axis equal off
    view(3)
    hold off
end
toc

```

EK B: GEN_FITNESS.m

```
function fitness = GEN_FITNESS(CL,CD_i,CM)
    global gg tail
    fitness = 1/CL/CD_i;
    if CL < gg
        fitness = 0;
    end
    if isempty(tail) == 0
        fitness = fitness/abs(CM(2));
    end
    if isnan(fitness) == 1
        fitness = 0;
    end
end
```

EK C: GEN_NEWBORN.m

```
function newborn = GEN_NEWBORN
    global specimen ns sequence constraint
    newborn = specimen;
    for s = 1:ns
        if sequence{s} ~= 0
            for n = 1:length(sequence{s})
                newborn{s}{sequence{s}(n)} = ...
                    constraint{s}{n}(1)+(constraint{s}{n}(2)-
constraint{s}{n}(1))*rand(1);
            end
        end
    end
end
```

EK D: GEN_PHENOTYPE.m

```
function GEN_PHENOTYPE(specimen,LineType)
    global ns
    for s = 1:ns
        P = GEN_PHENOTYPE_PREPROCESS(specimen{s});
        [ni1,nj1] = size(P);
        ni = ni1-1;
        nj = nj1-1;
        hold on
        for i = 1:ni
            for j = 1:nj
                plot3([P{i,j}(1) P{i+1,j}(1)],...
                    [P{i,j}(2) P{i+1,j}(2)],...
                    [P{i,j}(3) P{i+1,j}(3)],LineType)
                plot3([P{i,j}(1) P{i,j+1}(1)],...
                    [P{i,j}(2) P{i,j+1}(2)],...
                    [P{i,j}(3) P{i,j+1}(3)],LineType)
                plot3([P{end,j}(1) P{end,j+1}(1)],...
                    [P{end,j}(2) P{end,j+1}(2)],...
                    [P{end,j}(3) P{end,j+1}(3)],LineType)
            end
            plot3([P{i,end}(1) P{i+1,end}(1)],...
                [P{i,end}(2) P{i+1,end}(2)],...
```

```

                                [P{i,end}(3) P{i+1,end}(3)],LineType)
    end
end
end

```

EK E: GEN_PHENOTYPE_PREPROCESS.m

```

function P = GEN_PHENOTYPE_PREPROCESS(surface)
    % Importing Surface
    S = surface{02};
    AR = surface{03};
    TR = surface{04};
    a_in = surface{05};
    a_tw = surface{06};
    a_sw = surface{07};
    a_di = surface{08};
    H = surface{09};
    R = surface{10};
    ni = 1;
    nj = 1;
    ss = surface{13};
    % Processing Input
    b = sqrt(AR*S);
    c_bar = S/b;
    c_0 = 3/2*c_bar*(1+TR)/(1+TR+TR^2);
    c_1 = c_0*TR;
    % Generating Camber Lines
    camber = [linspace(0,1,ni+1)' zeros(ni+1,1)];
    camber = [camber(:,1) zeros(ni+1,1) camber(:,2)];
    % Generating Panel Ends
    P = cell(ni+1,nj+1);
    if ss == 0
        s = b;
    else
        s = b/2;
    end
    dx = s*tand(a_sw)/nj;
    dy = s/nj;
    m = (c_1-c_0)/s;
    for i = 1:ni+1
        for j = 1:nj+1
            % Baseline
            p = camber(i,:);
            % Sizing
            y = dy*(j-1);
            c = m*y+c_0;
            p = c*p;
            % Positioning
            p = p-H*c*[1;0;0];
            % Twisting
            a = a_in+a_tw*y/s;
            r = [cosd(-a) 0 -sind(-a);0 1 0;sind(-a) 0 cosd(-a)];
            p = r*p;
            % Translating
            x = dx*(j-1);
            p = p+x*[1;0;0];
            p = p+y*[0;1;0];
            % Rotating
            a = a_di;
            if y > 0

```

```

a)];
        r = [1 0 0;0 cosd(-a) sind(-a);0 -sind(-a) cosd(-
        p = r*p;
        end
        % Translating
        p = p+R;
        % Panel Ends
        P{i,j} = p;
    end
end
% Symmetry
if ss == 1
    ysym = [+1 0 0
            0 -1 0
            0 0 +1];
    P0 = P;
    P = cell(ni+1,2*nj+1);
    for i = 1:ni+1
        count = 0;
        for j = 1:2*nj+1
            count = count+1;
            if count <= nj
                P{i,j} = ysym*P0{i,nj+1-count+1};
            else
                P{i,j} = P0{i,count-nj};
            end
        end
    end
end
end
end
end

```

EK F: GEN_SELECTION.m

```

function [POP,FIT] = GEN_SELECTION(POP,FIT)
global np mr mod
for p = 1:np
    NORM = GEN_SELECTION_NORMALIZE(FIT);
    SUM = sum(NORM);
    pool = [];
    for i = 1:np
        dominance = round((NORM(i)/SUM)*100);
        for j = 1:dominance
            pool = [pool;i];
        end
    end
    if length(unique(pool)) > 1
        parent_1 = POP{p};
        partner = p;
        while partner == p
            partner = datasample(pool,1);
            parent_2 = POP{partner};
        end
        child = GEN_SELECTION_CROSSOVER(parent_1,parent_2);
        [CL,CD_i,CM] = VLM(child);
        fitness_child = GEN_FITNESS(CL,CD_i,CM);
        if rand <= mr
            child = GEN_SELECTION_MUTATION(child);
            [CL,CD_i,CM] = VLM(child);
            fitness_child = GEN_FITNESS(CL,CD_i,CM);
        end
    end
end

```

```

end
if mod == 1
    POP{p} = child;
    FIT(p) = fitness_child;
elseif mod == 2
    if fitness_child > FIT(p)
        POP{p} = child;
        FIT(p) = fitness_child;
    end
elseif mod == 3 && fitness_child > min(FIT)
    id_weak = find(FIT==min(FIT));
    if length(id_weak) > 1
        id_weak = datasample(id_weak,1);
    end
    POP{id_weak} = child;
    FIT(id_weak) = fitness_child;
end
end
end
end
end

```

EK G: GEN_SELECTION_CROSSOVER.m

```

function child = GEN_SELECTION_CROSSOVER(parent_1,parent_2)
    global specimen ns sequence
    child = specimen;
    for s = 1:ns
        if sequence{s} == 0
            break
        else
            nn = length(sequence{s});
            id = ones(nn,1);
            while isequal(id,ones(nn,1)) == 1 ||
isequal(id,2*ones(nn,1)) == 1
                for n = 1:nn
                    id(n) = randi([1 2]);
                end
            end
            for n = 1:nn
                if id(n) == 1
                    child{s}{sequence{s}(n)} =
parent_1{s}{sequence{s}(n)};
                else
                    child{s}{sequence{s}(n)} =
parent_2{s}{sequence{s}(n)};
                end
            end
        end
    end
end
end
end

```

EK H: GEN_SELECTION_MUTATION.m

```

function mutant = GEN_SELECTION_MUTATION(child)
    global ns sequence constraint
    mutant = child;
    for s = 1:ns

```

```

    if sequence{s} == 0
        break
    else
        if randi([0 1]) == 1
            for n = 1:length(sequence{s})
                mutant{s}{sequence{s}(n)} = ...
                    constraint{s}{n}(1)+(constraint{s}{n}(2)-
constraint{s}{n}(1))*rand(1);
            end
        end
    end
end
end
end
end

```

EK I: GEN_SELECTION_NORMALIZE.m

```

function [NORM] = GEN_SELECTION_NORMALIZE(FIT)
    global np
    MIN = min(FIT);
    MAX = max(FIT);
    NORM = zeros(np,1);
    for p = 1:np
        NORM(p) = (FIT(p)-MIN)/MAX;
    end
end

```

EK I: VLM.m

```

function [CL,CD_i,CM] = VLM(specimen)
    global ns S c_bar
    ns = size(specimen,1);
    S = zeros(ns,1);
    C = cell(ns,1);
    N = cell(ns,1);
    V = cell(ns,1);
    B = cell(ns,1);
    c_bar = zeros(ns,1);
    for s = 1:ns
        [S(s),c_bar(s),C{s},N{s},V{s},B{s}] =
VLM_PREPROCESS(specimen{s});
    end
    gamma = VLM_PROCESS(C,N,V);
    [CL,CD_i,CM] = VLM_POSTPROCESS(V,B,gamma);
end

```

EK J: VLM_BIOTSAVART.m

```

function v = VLM_BIOTSAVART(c,v1,v2)
    r0 = v2-v1;
    r1 = c-v1; d1 = norm(r1);
    r2 = c-v2; d2 = norm(r2);
    eps = 1e-6;
    if norm(r1)<eps || norm(r2)<eps || norm(cross(r1,r2))<eps
        v = [0 0 0]';
    end

```

```

else
    r1xr2 = cross(r1,r2);
    v = (1/4/pi)*(r1xr2/norm(r1xr2)^2)*dot(r0,(r1/d1)-(r2/d2));
end
end

```

EK K: VLM_POSTPROCESS.m

```

function [CL,CD_i,CM] = VLM_POSTPROCESS(V,B,gamma)
global ns ref U_inf V_inf rho_inf q_inf S c_bar
ni = zeros(ns,1);
nj = zeros(ns,1);
V_dw = cell(ns,1);
F = cell(ns,1);
e_l = cross(1/U_inf*V_inf,[0;1;0]);
e_d = 1/U_inf*V_inf;
l = zeros(ns,1);
Cl = zeros(ns,1);
d_i = zeros(ns,1);
Cd_i = zeros(ns,1);
m = cell(1,ns);
Cm = cell(1,ns);
for s = 1:ns
    [ni(s),nj(s)] = size(B{s});
    V_dw{s} = cell(ni(s),nj(s));
end
for sC = 1:ns
    F{sC} = zeros(3,1);
    m{sC} = zeros(3,1);
    Cm{sC} = zeros(3,1);
    for iC = 1:ni(sC)
        for jC = 1:nj(sC)
            b = B{sC}{iC,jC};
            v = 0;
            for sV = 1:ns
                for iV = 1:ni(sV)
                    for jV = 1:nj(sV)
                        nt = size(V{sV},1)-1;
                        for tV = nt:-1:iV
                            v1 = V{sV}{tV+1,jV};
                            v2 = V{sV}{tV,jV};
                            v =
v+gamma{sV}(iV,jV)*VLM_BIOTSAVART(b,v1,v2);
                        end
                            v1 = V{sV}{iV,jV};
                            v2 = V{sV}{iV,jV+1};
                            v =
v+gamma{sV}(iV,jV)*VLM_BIOTSAVART(b,v1,v2);
                        for tV = iV:nt
                            v1 = V{sV}{tV,jV+1};
                            v2 = V{sV}{tV+1,jV+1};
                            v =
v+gamma{sV}(iV,jV)*VLM_BIOTSAVART(b,v1,v2);
                        end
                    end
                end
            end
        end
    end
    V_dw{sC}{iC,jC} = v;
    V_eff = V_inf+V_dw{sC}{iC,jC};

```

```

        f =
cross(rho_inf*V_eff, gamma{sC}(iC, jC) * (V{sC}{iV, jV+1} - V{sC}{iV, jV}));
        F{sC} = F{sC} + f;
        m{sC} = m{sC} + cross(B{sC}{iC, jC}, f);
    end
end
l(sC) = dot(F{sC}, e_l);
d_i(sC) = dot(F{sC}, e_d);
Cl(sC) = l(sC) / q_inf / S(sC);
Cd_i(sC) = d_i(sC) / q_inf / S(sC);
Cm{sC} = m{sC} / q_inf / S(sC) / c_bar(sC);
end
L = sum(l);
D_i = sum(d_i);
M = sum(cell2mat(m), 2);
CL = L / q_inf / sum(S(ref));
CD_i = D_i / q_inf / sum(S(ref));
CM = M / q_inf / sum(S(ref)) / c_bar(ref(1));
end

```

EK L: VLM_PREPROCESS.m

```

function [S, c_bar, C, N, V, B] = VLM_PREPROCESS(surface)
% Freestream Properties
global alpha scl
% Importing Surface
profile = surface{01};
S = surface{02};
AR = surface{03};
TR = surface{04};
a_in = surface{05};
a_tw = surface{06};
a_sw = surface{07};
a_di = surface{08};
H = surface{09};
R = surface{10};
ni = surface{11};
nj = surface{12};
ss = surface{13};
% Processing Input
if ss == 1
    if mod(nj, 2) == 0
        nj = nj / 2;
    else
        error('nj must be even for symmetrical surfaces.')
    end
end
b = sqrt(S * AR);
c_bar = b / AR;
c_0 = 1.5 * c_bar * (1 + TR) / (1 + TR + TR^2);
c_1 = c_0 * TR;
% Generating Camber Lines
if ischar(profile) == 1
    if strcmp(profile, 'FLAT') == 1
        camber = [linspace(0, 1, ni + 1) zeros(ni + 1, 1)];
    elseif strcmp(profile, 'LOAD') == 1
        camber = VLM_PREPROCESS_LOAD(ni);
    else
        camber = VLM_PREPROCESS_NACA(profile, ni);
    end
end

```



```

end
else
    error('Profile not recognized.')
end
camber = [camber(:,1) zeros(ni+1,1) camber(:,2)];
% Generating Panel Ends
P = cell(ni+1,nj+1);
if ss == 0
    s = b;
else
    s = b/2;
end
dx = s*tand(a_sw)/nj;
dy = s/nj;
m = (c_1-c_0)/s;
for i = 1:ni+1
    for j = 1:nj+1
        % Baseline
        p = camber(i,:);
        % Sizing
        y = dy*(j-1);
        c_bar = m*y+c_0;
        p = c_bar*p;
        % Positioning
        p = p-H*c_bar*[1;0;0];
        % Twisting
        a = a_in+a_tw*y/s;
        r = [cosd(a) 0 sind(a)
            0 1 0
            -sind(a) 0 cosd(a)];
        p = r*p;
        % Translating
        x = dx*(j-1);
        p = p+x*[1;0;0];
        p = p+y*[0;1;0];
        % Rotating
        a = a_di;
        if ss == 1
            if y > 0
                r = [1 0 0
                    0 cosd(a) -sind(a)
                    0 sind(a) cosd(a)];
                p = r*p;
            end
        else
            r = [1 0 0
                0 cosd(a) -sind(a)
                0 sind(a) cosd(a)];
            p = r*p;
        end
        % Translating
        p = p+R;
        % Panel Ends
        P{i,j} = p;
    end
end
% Generating Control Points
C = cell(ni,nj);
for i = 1:ni
    for j = 1:nj
        C1 = (P{i,j}+P{i,j+1})/2;
    end
end

```

```

        C2 = (P{i+1,j}+P{i+1,j+1})/2;
        C{i,j} = C1+3/4*(C2-C1);
    end
end
% Panel Normals
N = cell(ni,nj);
for i = 1:ni
    for j = 1:nj
        d1 = P{i+1,j+1}-P{i,j};
        d2 = P{i,j+1}-P{i+1,j};
        N{i,j} = cross(d1,d2)/norm(cross(d1,d2));
    end
end
% Generating Vortex Ends
V = cell(ni+2,nj+1);
for i = 1:ni+1
    for j = 1:nj+1
        if i == ni+1
            V{i,j} = P{i,j};
        else
            v = P{i+1,j}-P{i,j};
            V{i,j} = P{i,j}+1/4*v;
        end
    end
end
scl = 100;
for j = 1:nj+1
    V{end,j} = P{end,j}+scl*b*[cosd(alpha);0;sind(alpha)];
end
% Generating Equivalent Bound Vortices
B = cell(ni,nj);
for i = 1:ni
    for j = 1:nj
        B{i,j} = 1/2*(V{i,j}+V{i,j+1});
    end
end
% Symmetry
if ss == 1
    ysym = [+1 0 0
            0 -1 0
            0 0 +1];
    P0 = P;
    C0 = C;
    N0 = N;
    V0 = V;
    B0 = B;
    P = cell(ni+1,2*nj+1);
    C = cell(ni,2*nj);
    N = cell(ni,2*nj);
    V = cell(ni+2,2*nj+1);
    B = cell(ni,2*nj);
    for i = 1:ni+1
        count = 0;
        for j = 1:2*nj+1
            count = count+1;
            if count <= nj
                P{i,j} = ysym*P0{i,nj+1-count+1};
            else
                P{i,j} = P0{i,count-nj};
            end
        end
    end
end

```

```

end
for i = 1:ni
    count = 0;
    for j = 1:2*nj
        count = count+1;
        if count <= nj
            C{i,j} = ysym*C0{i,nj-count+1};
            N{i,j} = ysym*N0{i,nj-count+1};
            B{i,j} = ysym*B0{i,nj-count+1};
        else
            C{i,j} = C0{i,count-nj};
            N{i,j} = N0{i,count-nj};
            B{i,j} = B0{i,count-nj};
        end
    end
end
for i = 1:ni+2
    count = 0;
    for j = 1:2*nj+1
        count = count+1;
        if count <= nj
            V{i,j} = ysym*V0{i,nj+1-count+1};
        else
            V{i,j} = V0{i,count-nj};
        end
    end
end
end
end
end

```

EK M: VLM_PREPROCESS_LOAD.m

```

function camber = VLM_PREPROCESS_LOAD(ni)
    profile = load('profile.dat');
    n = size(profile,1);
    x = 0.5*(flipud(profile(1:(n+1)/2,1))+profile((n+1)/2:n,1));
    y = 0.5*(flipud(profile(1:(n+1)/2,2))+profile((n+1)/2:n,2));
    f = fit(x,y,'cubicinterp');
    n = ni+1;
    a = linspace(pi,0,n)';
    xc = (cos(a)+1)/2;
    yc = f(xc);
    camber = [xc,yc];
end

```

EK N: VLM_PREPROCESS_NACA.m

```

function camber = VLM_PREPROCESS_NACA(profile,ni)
    digits = numel(profile);
    n = ni+1;
    a = linspace(pi,0,n)';
    xc = (cos(a)+1)/2;
    if digits == 4
        XX = str2num(profile(3:4));
        P = str2num(profile(2));
    end
end

```

```

M = str2num(profile(1));
m = M/100;
p = P/10;
y4f = @(x) (m/(p^2))*(2*p*x-x^2);
y4b = @(x) (m/(1-p)^2)*(1-2*p+2*p*x-x^2);
yc = zeros(n,1);
for i = 1:n
    if xc(i) < p
        yc(i) = y4f(xc(i));
    elseif xc(i) >= p
        yc(i) = y4b(xc(i));
    end
end
camber = [xc,yc];
elseif digits == 6 && profile(1) == '6' && profile(3) == '-'
c = 0.1*str2num(profile(end-2));
A = 0.8;
g = -1/(1-A)*(A^2*(0.5*log(A)-0.25)+0.25);
h = 1/(1-A)*(0.5*(1-A)^2*log(1-A)-0.25*(1-A)^2)+g;
y6 = @(x) c/(2*pi*(A+1))*(1/(1-A)*(...
    0.5*(A-x)^2*log2(abs(A-x))-0.5*(1-x)^2*log(1-
x)+...
    0.25*(1-x)^2-0.25*(abs(A-x))^2)-x*log(x)+g-h*x);
yc = zeros(n,1);
for i = 1:n
    yc(i) = y6(xc(i));
    if isnan(yc(i)) == 1
        yc(i) = 0;
    end
end
camber = [xc,yc];
else
error('Profile is not recognized.')
end
end
end

```

EK O: VLM_PROCESS.m

```

function gamma = VLM_PROCESS(C,N,V)
global ns V_inf
ni = zeros(ns,1);
nj = zeros(ns,1);
np = zeros(ns,1);
for s = 1:ns
    ni(s) = size(C{s},1);
    nj(s) = size(C{s},2);
    np(s) = ni(s)*nj(s);
end
A = zeros(sum(np));
R = zeros(sum(np),1);
i = 0;
for sC = 1:ns
    for iC = 1:ni(sC)
        for jC = 1:nj(sC)
            i = i+1;
            j = 0;
            c = C{sC}{iC,jC};
            n = N{sC}{iC,jC};
            for sV = 1:ns

```

```

[Vb,Vt] = VLM_PROCESS_VIND(c,V{sV});
for iV = 1:ni(sV)
    for jV = 1:nj(sV)
        j = j+1;
        vB = [Vb(iV,jV,1)
              Vb(iV,jV,2)
              Vb(iV,jV,3)];
        vL = [sum(Vt(iV:end,jV,1))
              sum(Vt(iV:end,jV,2))
              sum(Vt(iV:end,jV,3))];
        vR = [sum(Vt(iV:end,jV+1,1))
              sum(Vt(iV:end,jV+1,2))
              sum(Vt(iV:end,jV+1,3))];
        v = vB-vL+vR;
        A(i,j) = dot(n,v);
    end
end
end
R(i) = -dot(n,V_inf);
end
end
end
G = linsolve(A,R);
gamma = cell(ns,1);
i = 0;
for sB = 1:ns
    [ni2,nj1] = size(V{sB});
    ni(sB) = ni2-2;
    nj(sB) = nj1-1;
    gamma{sB} = zeros(ni(sB),nj(sB));
    for iB = 1:ni(sB)
        for jB = 1:nj(sB)
            i = i+1;
            gamma{sB}(iB,jB) = G(i);
        end
    end
end
end
end
end

```

EK Ö: VLM_PROCESS_VIND.m

```

function [Vb,Vt] = VLM_PROCESS_VIND(c,V)
[ni2,nj1] = size(V);
ni = ni2-2;
nj = nj1-1;
Vb = zeros(ni+1,nj,3);
for i = 1:ni+1
    for j = 1:nj
        v1 = V{i,j};
        v2 = V{i,j+1};
        v = VLM_BIOTSAVART(c,v1,v2);
        Vb(i,j,1) = v(1);
        Vb(i,j,2) = v(2);
        Vb(i,j,3) = v(3);
    end
end
Vt = zeros(ni+1,nj+1,3);
for i = 1:ni+1
    for j = 1:nj+1

```

```
v1 = V{i,j};  
v2 = V{i+1,j};  
v = VLM_BIOTSAVART(c,v1,v2);  
Vt(i,j,1) = v(1);  
Vt(i,j,2) = v(2);  
Vt(i,j,3) = v(3);  
    end  
end  
end
```



ÖZGEÇMİŞ



Ad-Soyad : Osman Mirza Demircan
Doğum Tarihi ve Yeri : 20 Haziran 1992, Altındağ/ANKARA
E-posta : demircano16@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** : 2016, Türk Hava Kurumu Üniversitesi, Havacılık ve Uzay Bilimleri Fakültesi, Uçak Mühendisliği
- **Yüksek Lisans** : 2018, İstanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri, Uçak ve Uzay Mühendisliği

MESLEKİ DENEYİM:

- Mart 2017'den beri Anatek Teknoloji Transfer Sanayi ve Ticaret A.Ş.'de Ar-Ge mühendisi olarak çalışmaktadır.