T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PALETLİ POMPALARDA KANAT SAYISI VE DÖNME HIZI DEĞİŞİMLERİNİN POMPA PERFORMANSINA ETKİLERİNİN İNCELENMESİ

MEHMET EMİN ÇETİN

YÜKSEK LİSANS TEZİ
MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI
KONSTRÜKSİYON PROGRAMI

DANIŞMAN
YRD. DOÇ. DR. MUHARREM E. BOĞOÇLU

İSTANBUL, 2015

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

PALETLİ POMPALARDA KANAT SAYISI VE DÖNME HIZI DEĞİŞİMLERİNİN POMPA PERFORMANSINA ETKİLERİNİN İNCELENMESİ

Mehmet Emin ÇETİN tarafından hazırlanan tez çalışması 14.05.2015 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Muharrem E. BOĞOÇLU

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Muharrem E. BOĞOÇLU

Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. İlyas İSTİF

Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. İsmail GERDEMELİ

İstanbul Teknik Üniversitesi

ÖNSÖZ

Akışkan akımının incelendiği, paletli pompalar ile ilgili hazırlanan bu çalışmanın pompa tasarımına önemli katkılar sağlamasını umuyorum.

Tez çalışmam boyunca beni yönlendiren ve destekleyen Sayın Yrd. Doç. Dr. Muharrem E. BOĞOÇLU hocama, bu çalışmaya olan katkılarından dolayı Sn. Arş. Gör. Aydın DÖNMEZ, Sn. Şule ÖZKAN ve Sn. Alper TOMBAK 'a teşekkür ederim. Her zaman olduğu gibi bu çalışmamda da bana desteklerini esirgemeyen çok kıymetli aileme şükranlarımı sunarım.

Mart, 2015

Mehmet Emin ÇETİN

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vi

KISALTMA LİSTESİ vii

ŞEKİL LİSTESİ viii

ÇİZELGE LİSTESİ x

ÖZET xi

ABSTRACT xiii

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 1
1.2 Tezin Amacı 3
1.3 Hipotez 3

BÖLÜM 2

PALETLİ POMPALAR 4

2.1 Çalışma Prensibi 4
2.2 Palet Tipleri 6
2.3 Malzeme Seçimi 7
2.4 Karşılaşılan Problemler 7
2.4.1 Aşınma 7
2.4.2 Dengelenmemiş Kuvvet 7
2.4.3 Kavitasyon 8
2.5 Değişken Deplasmanlı Paletli Pompalar 9

BÖLÜM 3

HESAPLAMALI AKIŞKANLAR DİNAMİĞİ 11

3.1 Akışkan Hareketi Temel Denklemleri 12
3.2 HAD Ayrıklaştırma Yöntemleri 13
3.2.1 Sonlu Farklar Yöntemi 13

3.2.2 Sonlu Elemanlar Yöntemi 14
3.2.3 Sonlu Hacimler Yöntemi 15
3.3 HAD Çözüm Adımları 15
3.3.1 Ağ Oluşturulması 16
3.3.2 Sınır Şartları 18
3.3.3 Çözüm Modelinin Belirlenmesi 18
3.3.4 Son işlem (Postprocess) ve Doğrulama 19

BÖLÜM 4

PALETLİ POMPA HAD ANALİZİ 20

4.1 Geometrinin Oluşturulması ve Ağ (Mesh) Örme 21
4.2 Sınır Şartları ve Akışkan Özellikleri 25
4.3 Kullanıcı Tanımlı Fonksiyon (User Defined Function - UDF) 27
4.4 Çözüm Modeli ve Metodunun Belirlenmesi 28
4.5 Analiz Sonuçları 30
4.5.1 Kanat Sayısına Göre Sonuçların İncelenmesi 30
4.5.2 Dönme Hızına Göre Sonuçların İncelenmesi 32
4.5.3 Pompa Çıkış Kanalı Boyutlarının Değiştirilmesinin Etkisi 34
4.5.4 Akışkan Akımı ile İlgili Sonuçlar 36

BÖLÜM 5

SONUÇ VE ÖNERİLER 43

KAYNAKLAR 45

EK-A

8 Kanatlı Pompa Sonuçları 47

EK-B

14 Kanatlı Pompa Sonuçları 49

EK-C

17 Kanatlı Pompa Sonuçları 51

EK-D

Çeşitli Kanat Sayılarına Göre Akım Çizgileri 53

EK-E

Kullanıcı Tanımlı Fonsiyon (UDF) Kodları 55

EK-F

SHELL SPİRAX S4 ATF HDX 70

ÖZGEÇMİŞ 71

SİMGE LİSTESİ

k	Türbülans kinetik enerjisi
ε	Türbülans yitim hızı
P	Basınç, Pa
P’	Değiştirilmiş basınç, Pa
V	Kontrol hacmi
V	Hız ve büyüklüğü, m/s
x	Kartezyen koordinat
y	Kartezyen koordinat
z	Kartezyen koordinat
u	Kartezyen koordinat sisteminde x doğrultusundaki hız
v	Kartezyen koordinat sisteminde y doğrultusundaki hız
w	Kartezyen koordinat sisteminde z doğrultusundaki hız

KISALTMA LİSTESİ

CFD	Hesaplamalı Akışkanlar Dinamiği (Computational Fluid Dynamics)
HAD	Hesaplamalı Akışkanlar Dinamiği
UDF	Kullanıcı Tanımlı Fonksiyon (User Defined Function)

ŞEKİL LİSTESİ

Sayfa

Şekil 2. 1 Paletli pompa ana elemanlar 5
Şekil 2. 2 Paletli pompa çalışma prensibi 5
Şekil 2. 3 Kanat tipleri [12] 6
Şekil 2. 4 Dengelenmemiş kuvvet [11] 8
Şekil 2. 5 Dengelenmiş paletli pompa [11] 8
Şekil 2. 6 Değişken deplasmanlı paletli pompa [11] 9
Şekil 2. 7 Otomatik kontrol edilen değişken deplasmanlı paletli pompa [2] 10
Şekil 3. 1 Bir akışkan akımının gerçek ve simülasyon görüntüsü [15] 11
Şekil 3. 2 Sonlu farklar yöntemi [15] 14
Şekil 3. 3 Sonlu elemanlar yöntemi [15] 14
Şekil 3. 4 Sonlu hacimler yöntemi [15] 15
Şekil 3. 5 Ağ oluşturmada kullanılan eleman tipleri [16] 17
Şekil 3. 6 Basit bir hesaplama bölgesi için sınır şartları 18
Şekil 3. 7 Çeşitli HAD analizi uygulamaları[15] 19
Şekil 4. 1 Paletli pompa patlatılmış resmi 21
Şekil 4. 2 Akışkanın pompa içinde doldurduğu hacim 22
Şekil 4. 3 Kanat ile Stator arasında oluşturulan geçit 23
Şekil 4. 4 Rotor-Stator bölgesi ağ örülmüş görünümü 23
Şekil 4. 5 Giriş kanalı ağ örülmüş görünümü 24
Şekil 4. 6 Ağ örülmüş paletli pompa 24
Şekil 4. 7 Arayüz (interface) tanımlaması (sarı: arayüz) 25
Şekil 4. 8 Hesaplama bölgesindeki bölge ve yüzey tanımlamaları 26
Şekil 4. 9 Problemin şeması [16] 28
Şekil 4. 10 11 kanatlı paletli pompanın (4500 d/d) analizi - artıklar (residuals) 29
Şekil 4. 11 Kanat sayısına göre debi değişimi 30
Şekil 4. 12 Kanat sayısına göre ortalama debi değişimi 30
Şekil 4. 13 Kanat sayısına göre pompa iç basınç değişimleri 31
Şekil 4. 14 Dönme hızına göre debi değişimi 32
Şekil 4. 15 Dönme hızına göre ortalama debi değişimi 32
Şekil 4. 16 Dönme hızına göre ortalama pompa iç basınç değişimi 33
Şekil 4. 17 Pompa giriş ve çıkış kanalı ölçüleri (1- Normal, 2- Değiştirilmiş) 34
Şekil 4. 18 Ölçüleri değiştirilmiş kanal geometrisinin debiye etkisi (5 kanatlı) 35
Şekil 4. 18 Ölçüleri değiştirilmiş kanal geometrisinin debiye etkisi (11 kanatlı) 35
Şekil 4. 20 Hız profili alınan kesit 36

Şekil 4. 21 Pompa havuzu kesitindeki hız profili 37
Şekil 4. 22 Pompa havuzundaki hız vektörleri 37
Şekil 4. 23 Giriş kanalı boyunca akım çizgileri 38
Şekil 4. 24 Çıkış kanalı boyunca akım çizgileri 38
Şekil 4. 25 Akışa ait akım çizgileri 39
Şekil 4. 26 Pompa genelinde basınç dağılmı 40
Şekil 4. 27 Akışa ait hız vektörleri 41
Şekil 4. 28 Kanat sayısına göre rotor-stator bölgesi basınç dağılımları 42

ÇİZELGE LİSTESİ

Sayfa

Çizelge 4. 1 Akışkan özellikleri……26
Çizelge 4. 2 Kanat sayısına göre pompa iç basınç değişimleri…….. 31
Çizelge 4. 3 Dönme hızına göre pompa iç basınç değişimleri…… 33

ÖZET

PALETLİ POMPALARDA KANAT SAYISI VE DÖNME HIZI DEĞİŞİMLERİNİN POMPA PERFORMANSINA ETKİLERİNİN İNCELENMESİ

Mehmet Emin ÇETİN

Makina Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Muharrem E. BOĞOÇLU

Paletli pompalar birçok alanda orta ve düşük viskoziteli sıvıların transferinde yaygın olarak kullanılan pozitif deplasmanlı hidrolik pompalardır. Temel olarak gövde, gövde içerisine yerleştirilmiş bir rotor ve rotor üzerine açılmış kanalların içinde bulunan paletlerden (kanat da denilebilir) oluşmaktadır. Gövde ekseni ve rotor ekseni arasında belirli oranda bir mesafe vardır. Rotorun dönme hareketi sonucu paletler merkezkaç kuvveti etkisiyle dışarı doğru gövde iç yüzeyine kadar savrulur. Böylece kanatlar pompa içerisinde önce artan sonra azalan bir hacmi süpürür. Basit gibi görünen bu çalışma mantığında birçok etken ortaya çıkmaktadır. Pompanın çalışması rotorun dönme haretiyle doğrudan ilgilidir. Çünkü kanat uçları ile gövde arasında oluşacak sızdırmazlık ve sürtünme kuvveti dönme hareketinin bir sonucudur. Pompa dönme hızının etkilediği en önemli değişken ise debidir. Debi ve akışkan akımının ana değişkenleri hız ile basınç dönme hızıyla direkt olarak ilgilidir. Bunların yanında pompa içi basınç ve hız değişimlerini etkileyebilecek bir diğer parametre de palet sayısıdır.

Bu çalışmada paletli pompalarda dönme hızı ve palet sayısının debiye ve pompa içi akışkan akımına olan etkisi araştırılmıştır. Otomobil direksiyon sisteminde kullanılacak bir paletli pompa ele alınarak hesaplamalı akışkanlar dinamiği (CFD) analizleri yapılmıştır. Analizler için Ansys Fluent 14.5 yazılımı kullanılmıştır. Zamana bağlı, kullanıcı tanımlı fonksiyonlar (UDF) kullanılarak sayısal çözümler yapılmıştır. Kullanıcı tanımlı fonksiyonlar C gibi bir programlama dilinde hazırlanıp Fluent ’e yüklenerek yazılan fonksiyona bağlı olarak çözüm elde etmeye yarar. Başka bir değişle mevcut

kodlara ek olarak bazı özel fonksiyonların kullanılmasına olanak sağlanmaktadır. Pompanın dönme hareketinin tanımlanması bu şekilde yapılmıştır. Farklı dönme hızlarında ve farklı kanat sayılarında analizler yapılarak sonuçlar irdelenmiştir.

Anahtar Kelimeler: Paletli Pompa, kanat sayısı, dönme hızı, pompa tasarımı, hesaplamalı akışkanlar dinamiği (CFD)

ABSTRACT

NUMERICAL ANALYSIS ON EFFECTS OF VANE NUMBER VARIATIONS ON PERFORMANCE OF A VANE PUMP WITH VARIOUS ROTATIONAL SPEEDS

Mehmet Emin ÇETİN

Department of Mechanical Engineering

MSc. Thesis

Adviser: Asst. Prof. Muharrem E. BOĞOÇLU

The vane pumps use in many areas of transferring medium and low viscosity liquids commonly. Basically, the vane pump composed of three main parts. These are stator, rotor and vanes. Stator can be defined as the pump body. Rotor is placed inside of the pump housing. Vanes are located in the channel opened onto the rotor. There is a distance between the stator axis and the rotor axis. The result of the rotational movement, vanes are thrown outward to the inner surface of the stator under the influence of centrifugal force. Thus, the vanes are swept a volume into the pump. Although the operating principle may seem simple, there are many factors affecting the performance of the pump. Operation of the pump is directly related to the rotational motion of the rotor. Because the sealing and friction force will occur between the vane tips and the inner surfaces of stator is a result of rotational movement. The important variable that affects the flow rate of the pump is rotational speed. Flow rate, fluid velocity and internal pressure are directly related to the rotational speed. The other parameter that can affect the internal pressure and velocity is the number of vanes.

In this study, the effects of the rotational speed and the number of vanes on the flow characteristics are investigated. CFD analyses of a vane pump to be used in automobile steering system have been made with Ansys Fluent 14.5. The models were analized transient and used UDF codes. UDF is a function that we program that can be

dynamically loaded with the Fluent solver to enhance the standard features of the code. Defining the rotational motion of the pump is made in this way. Different rotational speeds and various vane numbers of analyses were performed and the results are discussed.

Keywords: Vane pump, number of vanes, rotational speed, pump design, computational fluid dynamics (CFD)

BÖLÜM 1

GİRİŞ

Pompalar dünyada birçok alanda kullanılan akışkanların transferini sağlayan makinalardır. Birçok karakteristik özelliğine göre gruplandırılabilen pompaların oldukça fazla sayıda çeşidi mevcuttur. Paletli pompalar genel sınıflandırmada pozitif yerdeğiştirmeli (hacimsel-pozitif iletimli) sınıf içine dâhildir. Diğer hacimsel pompalar gibi akışkanı pompanın iç hacmini değiştirilerek iletir.

1.1 Literatür Özeti

Pompalarla ilgili yayınlanmış birçok yayın, yapılmış birçok araştırma olmasına rağmen paletli pompa tasarımına yönelik pek fazla kaynak bulunmaktadır. Genellikle paletli pompaların kontrolüne yönelik araştırmalar yapılmış, aşınma konusunda ve bir diğer konu olan pompa verimini artırma yollarıyla ilgili çalışmalar yapılmış.

Li [1] doktora tezinde, hidrolik bir sisteme uyarlanan paletli pompanın modellemesini ve kontrolünü yaparak verimini araştırmıştır.

Karmel [2] çalışmasında değişken deplasmanlı paletli pompa iç basınç dağılımlarının çözümlemesini yapmış, pompa mekanizması ve milinde oluşan iç kuvvetleri ve torku incelemiştir. Bu çalışmanın başlıca sebepleri olarak, değişken deplasmanlı paletli pompanın tasarım, kontrol ve yataklama konuları gösterilebilir.

Cavallari [3] doktora tezinde toplu parametre modelleri ile simule edilebilen değişken deplasmanlı paletli pompaların titreşim davranışlarını sunmaktadır. Döner parçaların titreşim analizleri 3 serbestlik dereceli düzlemsel model ile ortaya koyulmuştur.

Wnag [4] deplasman kontrollü eksenel pistonlu bir pompanın hidrolik bir sistemde valfsiz kontrol edilmesi üzerine çalışmıştır. Temelde uygun hidrolik devre ve kontrol algoritması geliştirmek hedeflenmiştir. Deneysel ve teorik çalışmalar ile sistem performansı tespit edilmiştir. Değişken deplasmanlı pompa ile valf kullanmadan sistemde yüksek enerji tasarrufu sağlandığı görülmüştür.

Petrović, [5] çalışmasında paletli pompaların geçici rejim davranışlarını ele almıştır. Çift odalı konsantrik paletli pompa yapısı kullanılarak, modelleme ve simulasyon çalışması yapılmıştır.

Giuffrida ve Lanzafame [6] çalışmalarında dengelenmiş paletli pompanın teorik olarak debi hesabını incelemiştir.

Elashmawy [7] çalışmasında pompa yağ tipinin pompa performansına ve pompanın ömrüne etkisini araştırmış, kanat hızının önemli bir parametre olduğu, kanat ile kam halka arasındaki yağ filmi tabakası kalınlığının sürtünme kuvvetine çok etkisinin olmadığı sonucuna varmıştır.

Inaguma [8] çalışmasında dengelenmiş paletli pompada mekanik verimi incelemiş, palet kalınlığının, palet tipinin, kanat ile kam halka arasındaki sürtünmenin verime etkilerini araştırmıştır. Sonuç olarak, sürtünme katsayısının azaltılabildiği palet tiplerinde mekanik verimin arttığı görülmüştür.

Pan ve Lei [9] çalışmalarında otomobil direksiyon sistemi için tasarlanan değişken deplasmanlı paletli bir pompanın modellemesini yaparak geliştirilmesi yönünde çalışmış, deneysel olarak uygulamalar yapmışlardır. Sonuçta, teorik hesaplamalar ile deneysel veriler örtüşmüş, direksiyon sistemindeki enerji kayıplarının azaltılabildiği görülmüştür.

Frendo vd. [10] çalışmalarında 5 ile 9 kanadı bulunan paletli pompanın çalışma şartları ve bazı dizayn parametrelerine etkileri incelenmiştir. Çalışmasında MSC.Adams programını kullanarak simülasyon yapılmış, deneysel çalışmalarla kıyas yapılmıştır. Farklı geometrik konfigürasyonlar altında kuvvet ve tork incelenmiştir.

1.2 Tezin Amacı

Otomotiv sektöründe otomatik vites ve direksiyon sistemlerinde tercih edilen paletli pompaların tasarımına yönelik son zamanlarda oldukça fazla araştırmalar yapılmaktadır. Üretim kabiliyetinin yükselmesi ve verimliliğin ön plana çıkmasıyla mevcut sistemlerin geliştirilmesine yönelik araştırmalar hız kazanmıştır. Bu çalışmada paletli pompalarda kanat sayısının ve dönme hızının pompa performansına etkisi üzerine bir araştırma yapılarak belirli kriterler ortaya çıkarmak amaçlanmıştır. Bilgisayar yardımıyla sayısal çözüm metodlarından yararlanılmıştır.

1.3 Hipotez

Paletli pompalar rotorun dönme hızıyla doğrudan ilgili olarak pompalama işlemini gerçekleştirmektedir. Dönme hızının az olması pompanın doğru çalışmamasına sebep olacaktır. Ayrıca yüksek dönme hızlarına dinamik etkilerden dolayı çıkılması zor olacaktır. Pompa tasarımcısı paletli pompalarda kanat sayısının ne kadar olması gerektiği yönünde bir bir soru ile karşılaşabilir. Pompanın kullanım alanının da göz önünde bulundurularak kanat sayısının belirlenmesi gerekecektir. Ancak özellikle otomotiv sektöründe vites ve direksiyon sistemlerinde kullanılan pompalar için kanat sayısı az olmamalıdır.

BÖLÜM 2

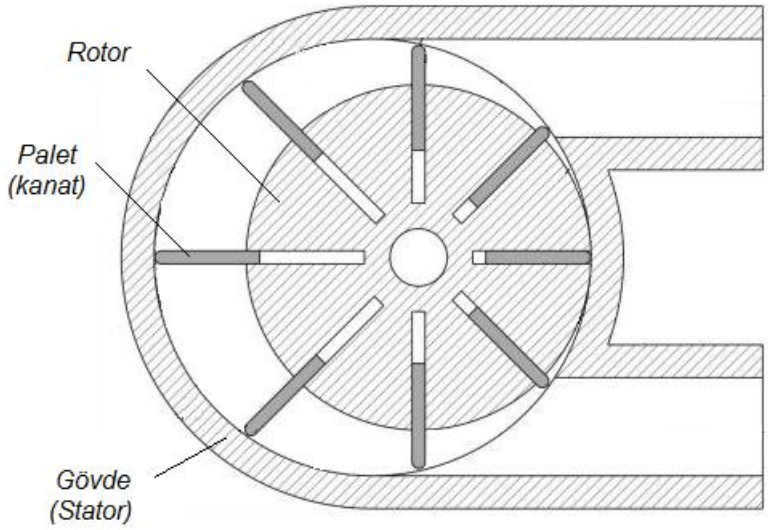
PALETLİ POMPALAR

Günümüzde birçok alanda kullanılan paletli pompalar orta ve yüksek basınç uygulamalarında sıkça kullanılmaktadır. Özelllikle otomotiv alanında otomatik vites, direksiyon sistemi uygulamalarında kullanılmaktadır. Havacılık servisi, petrol endüstrisi, yakıt transferleri, deniz taşımacılığı, güç üretimi gibi birçok alanda kullanımı uygun olup, LPG, amonyak, çözücüler, benzin, alkol ve fuel oil gibi düşük ve orta viskoziteli sıvıların transferinde idealdirler.

Paletli pompalar alçak gürültü seviyeleriyle ön planda olan bir hidrolik pompa türüdür. Pompa dış hacmine göre yüksek miktarlarda debi üretirler. Bunların yanısıra kompakt bir yapıda olmaları avantaj olarak görülebilir.

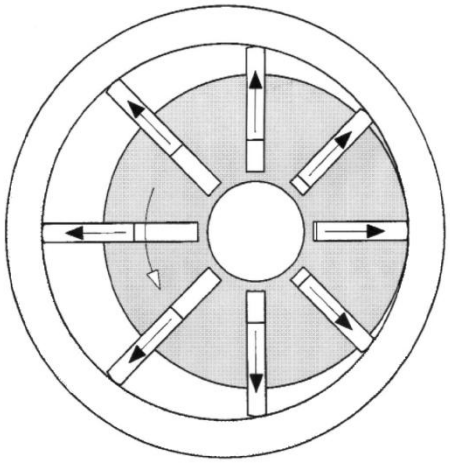
2.1 Çalışma Prensibi

Paletli pompalar, pompa gövdesi (stator) içine yerleştirilmiş dönebilen bir rotor ve rotor üzerine açılmış kanallarda içeri ve dışarı hareket edebilen kanatlar olmak üzere 3 temel bileşenden oluşur (Şekil 2.1). Gövde ekseni ile rotor merkezi arasında eksantriklik vardır. Pompa debisinin ayarlanmasında bu eksantriklik çok büyük bir etkendir. Merkezler arasındaki mesafe arttıkça pompa debisi de artmaktadır.



Şekil 2. 1 Paletli pompa ana elemanlar

Rotorun hareketi ile oluşan merkezkaç kuvvetinin etkisiyle kanatlar dışarı doğru savrulur. Merkezkaç kuvvetinin tam anlamıyla oluşup, sızdırmazlığın sağlanması için pompanın belirli bir hıza ulaşması şarttır. Kanatlı pompaların minimum devir sayısı 600 d/dak civarındadır [11]. Savrulan kanatlar rotor ile stator arasında belirli bir hacmi süpürmeye başlayacaktır. Bu hacim giriş ağzından çıkış ağzına doğru önce artar, sonra azalır. Bu sayade pompalama işlemi gerçekleşmektedir (Şekil 2.2).

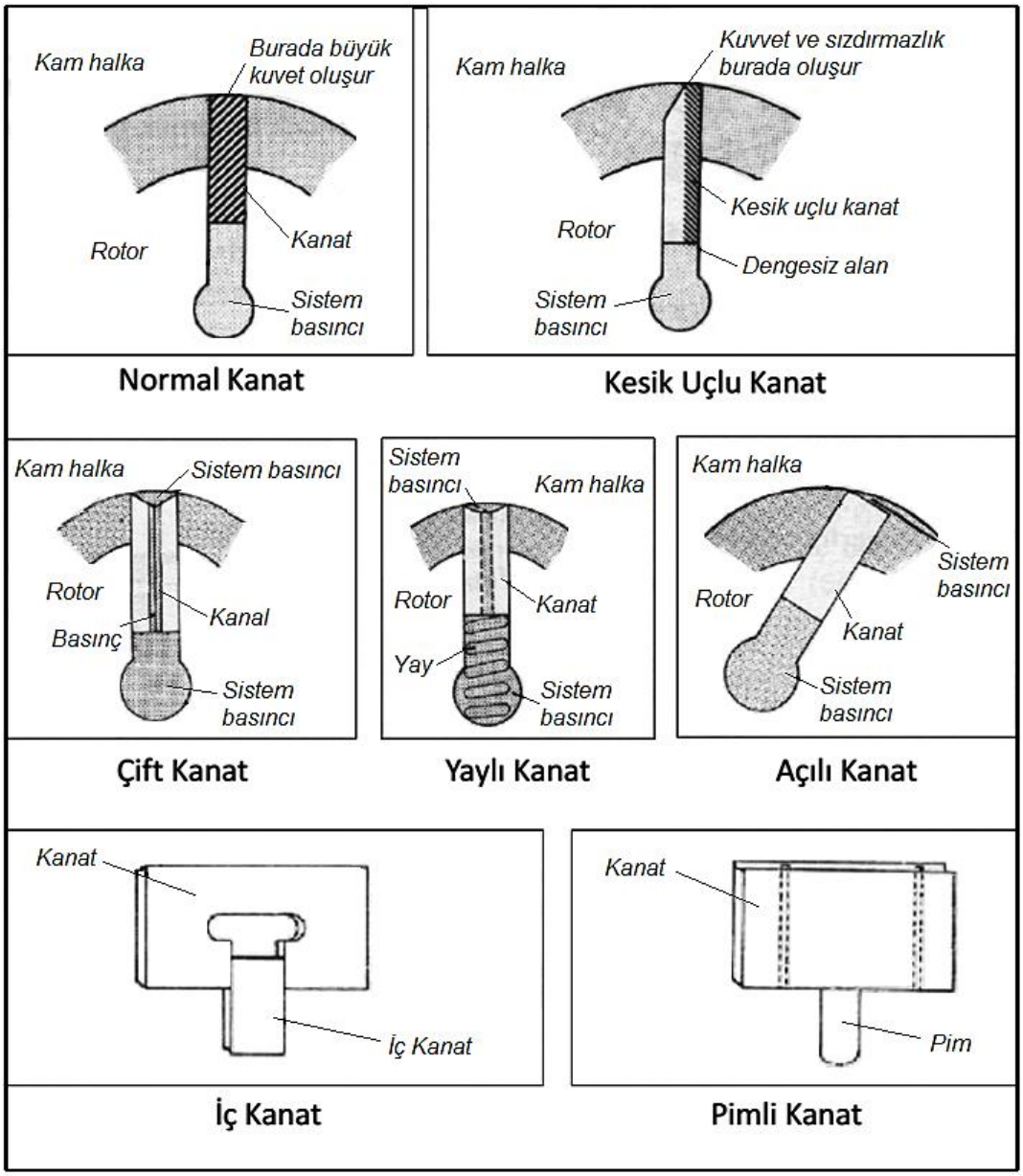


Şekil 2. 2 Paletli pompa çalışma prensibi

Ayrıca paletlerin dışarı doğru hareketi palet arkasına yerleştirilen yaylar ile veya sistem basıncıyla beslenmesiyle de desteklenebilmektedir. Bu tür uygulamalar sızdırmazlığı daha iyi sağlamak adına geliştirilmiştir.

2.2 Palet Tipleri

Farklı yapı ve tipte kanatlar kullanılmaktadır. Palet şeklinin belirlenmesindeki en önemli girdi palet ucunun pompa gövdesine (ya da aşınma halkasına) uygulayacağı kuvvetin kontrol edilmesidir. Bir diğer husus ise rotor kanalları içinde paletlerin sıkışmadan, serbest hareket edebilmesidir. Bütün bunların yanısıra palet seçiminde sızdırmazlık ön plana çıkmaktadır. Rotor kanalları ile paletler arası sızdırmazlık ve palet ucu ile aşınma halkası arasındaki sızdırmazlık yeterince sağlanmalıdır. Pompada kullanılacak sıvının viskozitesi de palet seçiminde göz önünde bulundurulmalıdır.



Şekil 2. 3 Kanat tipleri [12]

2.3 Malzeme Seçimi

Paletli pompa gövdeleri genellikle demir döküm veya çelik dökümden imal edilir. Rotor malzemesi olarak da genellikle çelik alaşımları kullanılır. Palet malzemesi seçiminde karbon grafit, viton, ryton, PPS, teflon, pik, yüksek hız çeliği gibi oldukça geniş bir yelpaze vardır. Kanat malzemesinin seçiminde pompa kullanım alanının önemi ön plana çıkmaktadır. Kullanılacak akışkanın özelliklerine bağlı olarak uygun palet malzemesi seçilmesi gerekmektedir.

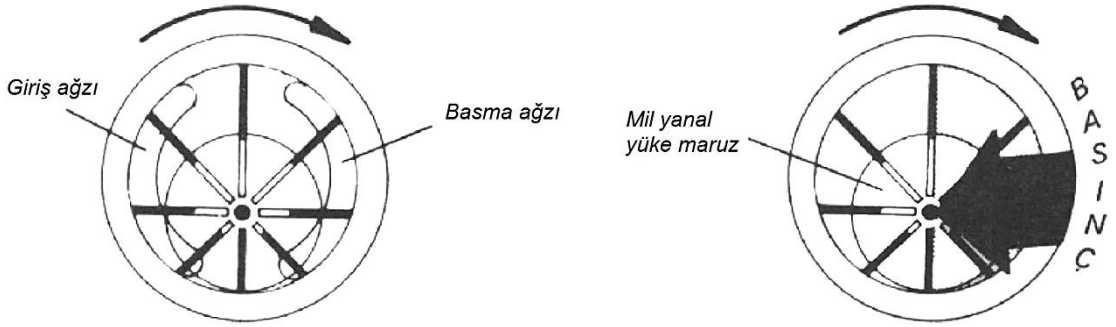
2.4 Karşılaşılan Problemler

2.4.1 Aşınma

Aşınma problemleri birçok alanda karşımıza çıktığı gibi paletli pompalarda da karşımıza çıkmaktadır. Pompa kanatları ile aşınma halkası arasındaki sürtünme pompa çalıştığı sürece devam etmektedir. Bu sürtünmeye bağlı olarak kanat uçlarında ve halkada aşınma meydana gelmektedir. Aşınma halkasının adı da buradan gelmektedir. Pompa gövdesi içine kanat uçlarının temas edeceği yere halka koyularak aşınma problemi sonucu halkanın değiştirilmesiyle pompanın kullanımına devam edilmesi amaçlanmaktadır. Palet tipi seçiminde aşınma ele alınması gereken önemli bir husustur.

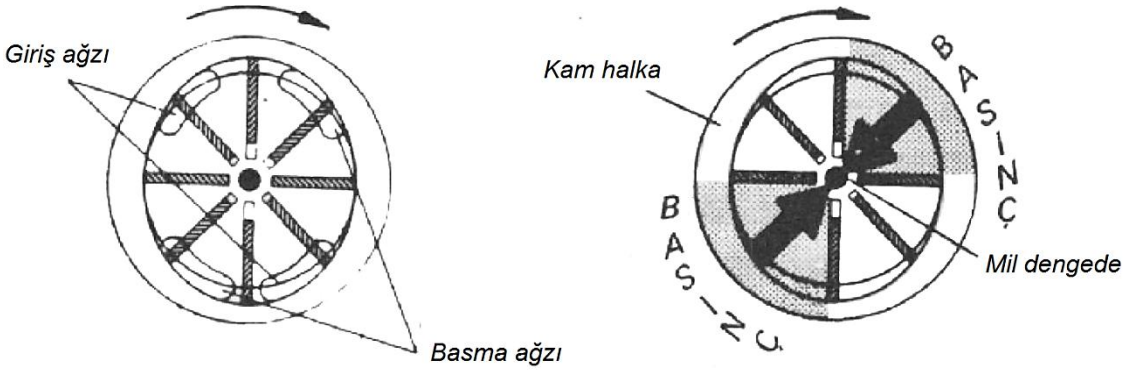
2.4.2 Dengelenmemiş Kuvvet

Paletli pompada çok farklı iki basınç vardır. Birincisi emiş bölümündeki atmosferik basınçtan düşük, diğeri basma ağzındaki çalışma basıncıdır. Özellikle çalışma basıncının yüksek olduğu durumlarda rotor mili tek yönde ciddi bir yanal yüke maruz kalır. Bundan dolayı bu tip pompalara dengelenmemiş kanatlı pompalar denir. Bu durumda (dengelenmemiş durum) oluşan yük, pompa mil yatağının aşınmasına neden olabilir. Bundan dolayı yüksek basınç gerektiren sistemlerde kullanılmazlar [11].



Şekil 2. 4 Dengelenmemiş kuvvet [11]

Pompa gövdesinde eliptik bir yapı kullanılarak, pompanın bir tam tur hareketinde iki pompalama yapılır. Böylece pompada iki giriş ve iki çıkış bölümü elde edilir. Sonuçta karşılıklı olarak birbirini dengeleyecek kuvvetler elde edilir. Bu tür pompalar da dengelenmiş (iki odalı) paletli pompa olarak isimlendirilir (Şekil 2.5).



Şekil 2. 5 Dengelenmiş paletli pompa [11]

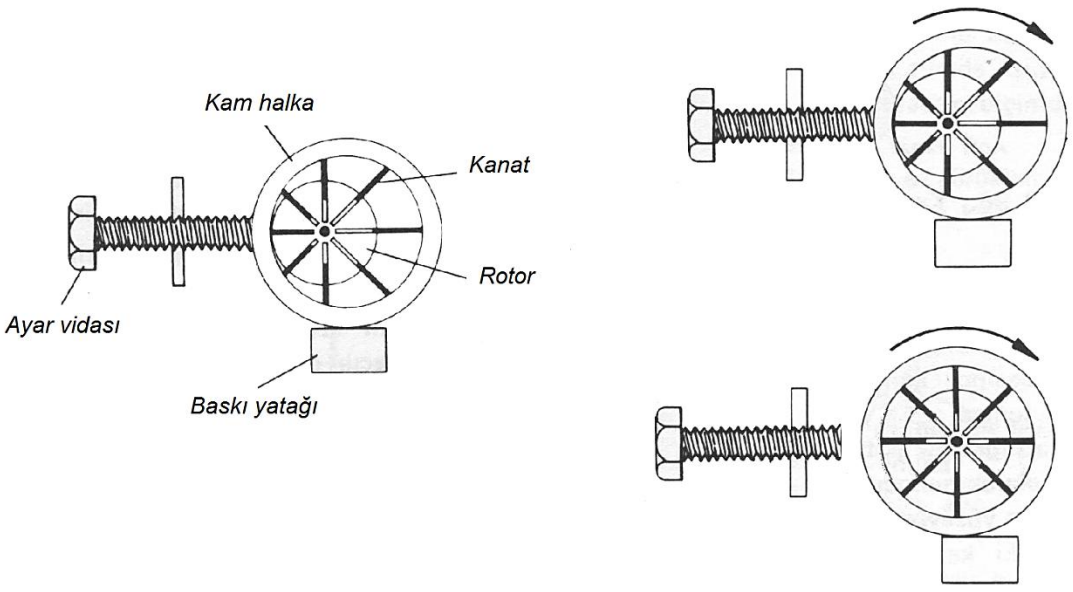
2.4.3 Kavitasyon

Pompa içinde herhangi bir bölgede sıvının statik basıncı buharlaşma basıncının altına düştüğü takdirde o bölgeden geçen sıvı buharlaşır ve çok sayısı küçük boyutlarda doymuş sıvı buharı kabarcıkları oluşur. Bununla birlikte sıvı içerisindeki hava veya gaz kabarcıkları oluşur. Akan sıvının pompa içinde çıkış ağzına doğru ilerlemesiyle sürüklenen kabarcıklar sıvının buharlaşma basıncının daha yüksek olduğu bir bölgeye geldiklerinde ise yoğuşarak tekrar sıvı haline dönerler. Bu şekilde basınç etkisiyle sıvının buharlaşıp yoğuşması olayına kavitasyon denir.

Pompada kavitasyon olayı gürültü, titreşim ve en önemlisi yoğuşma bölgelerinde kavitasyon erozyonuna neden olur. Erozyon sonucu malzeme aşınmasının sonucunda ise pompa veriminde düşüş meydana gelebilir.

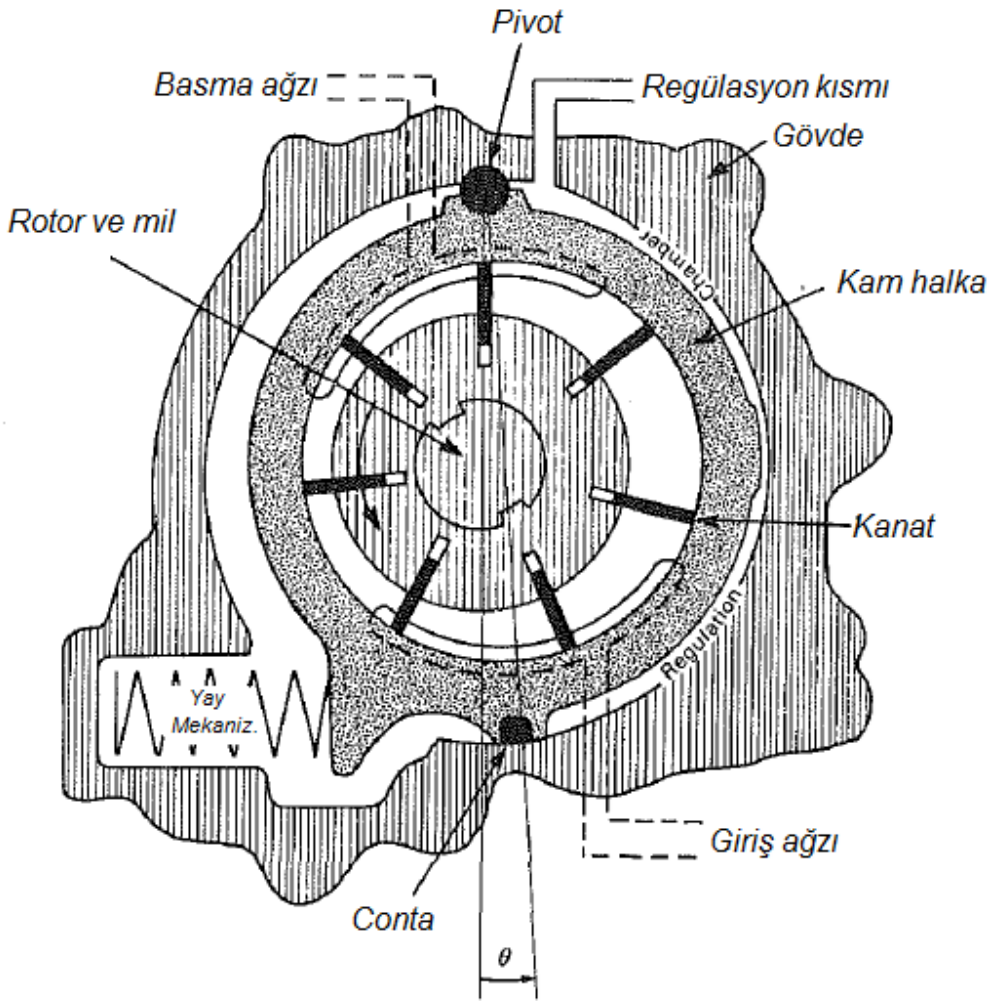
2.5 Değişken Deplasmanlı Paletli Pompalar

Günümüzde bir sistemin verimliliği en önemli konular arasındadır. Hidrolik bir sistemin çalışması sırasında sistem için gerekli olan yağ miktarı her zaman aynı miktarda olmamaktadır. Fakat sistemi besleyen pompalar genellikle sabit oranda yağı sisteme gönderirler. Bu da sistemin verimliliğini önemli ölçüde düşürmektedir. Paletli pompalarda debinin değiştirilmesi iki yolla sağlanabilir. Birincisi pompayı tahrik eden elektrik motorunun devir hızının değiştirilmesiyle, ikincisi ise pompanın basma hacminin değiştirilmesiyledir. Paletli pompanın basma hacmi kanat genişliği ve kanat stroğuna bağlıdır. Pompa çalışırken kanat genişliğini değiştirmek mümkün olmayacağından kanat stroğunu değiştirecek bir tasarım ihtiyacı ortaya çıkmaktadır.



Şekil 2. 6 Değişken deplasmanlı paletli pompa [11]

Basit olarak değişken deplasmanlı paletli bir pompa serbest hareket edebilen bir gövde, gövdeye kılavuzluk eden baskı yatağı ve gövdenin konumunu değiştiren ayar vidasından oluşur. Ayar vidası vasıtasıyla gövdenin hareketi sonucu basma hacmi değişecektir.



Şekil 2. 7 Otomatik kontrol edilen değişken deplasmanlı paletli pompa [2]

Değişken deplasmanlı paletli pompalarda rotor ve stator eksenleri arasındaki kaçıklığın değiştirilebileceği birçok yöntem vardır. Hidrolikle beslenen bir yapı ve sisteme uyarlanmış bir yay mekanizmasıyla eksantriklik kontrol edilebilir. Kontrol yöntemleri manuel olduğu gibi otomatik de olabilmektedir.

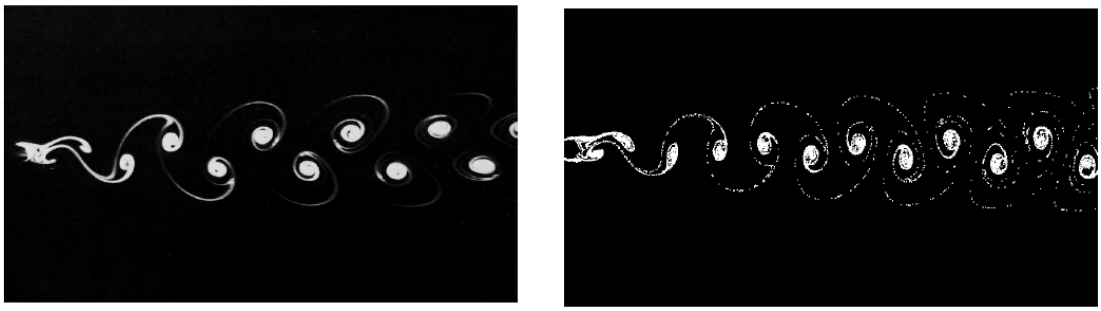
BÖLÜM 3

HESAPLAMALI AKIŞKANLAR DİNAMİĞİ

Akışın olduğu mühendislik sistemlerinin tasarım ve analizinde iki temel yaklaşım vardır: deney yapma ve hesaplama. Bunlardan ilki, tipik olarak rüzgâr tünelinde veya başka yerlerde test edilecek modellerin yapımını gerektirirken ikincisi diferansiyel denklemlerin analitik veya sayısal olarak çözümlemesini içerir [13].

Hesaplamalı akışkanlar dinamiği, akışkan akımı ile ilgili özellikleri tanımlayan temel kütle, momentum ve enerji denklemlerinin belirtilen sınır şartları içerisinde bilgisayar yazılımları yardımıyla çözülmesine dayanan sayısal bir yöntemdir. Bu yöntemle, üzerinde çalışılmak istenen sistemi temsil eden bir model oluşturulabilmekte ve akışkan akımı ile ilgili fiziksel ve kimyasal özellikler bu görsel model üzerine uygulanarak, kullanılan yazılım vasıtasıyla tanımlanan sistemle ilgili gerçeğe yakın tahminler yapılabilmektedir.

Bu bölümde hesaplamalı akışkanlar dinamiği hakkında bir fikir oluşması için kısa bir giriş yapılarak, HAD analizi hakkında yüzeysel bilgiler sunulmaktadır. Çalışmada Ansys FLUENT 14.5 yazılımı kullanılmış olup, konular bu yazılıma paralel olarak ele alınmıştır.



Şekil 3. 1 Bir akışkan akımının gerçek ve simülasyon görüntüsü [15]

3.1 Akışkan Hareketi Temel Denklemleri

Akışkanlar mekaniği temel denklemleri kütle, momentum ve enerjinin korunum kanunlarını esas alır. Kütlenin korunumu süreklilik denklemini olarak tanımlanır. Momentum denklemleri Newton 'un 2. kanununun akışkanlar mekaniğine uyarlanmasıdır ve transport denklemleri olarak isimlendirilir. Enerji denklemi ise termodinamiğin 1. yasasına dayanır.

Akışkanlar mekaniği denklemlerinin çıkarılmasında izlenen yöntem kontrol hacmi yaklaşımıdır. Kontrol hacmi akışkanın hareketli olduğu kontrol yüzeyleri ile sınırlandırılmış bir çözüm alanını ifade eder.

Süreklilik denklemi tek bir denklem ile ifade edilen skaler bir denklemken, momentum denklemleri x, y, z bileşenleri için ayrı ayrı ifade edilen vektörel formda bir denklemdir. Enerji denklemi ise skaler formda ifade edilir. Sonuç olarak akışkanlar mekaniği temel denklemleri kütle, x-y-z momentum, enerji ve hal denklemlerinden oluşur. Eğer akış problemi kimyasal olaylar içeriyorsa bu denklemlere ek bazı özel denklemlerin de hesaba katılması gerekmektedir. Problemin durumuna göre bahsi geçen denklemlerin hepsinin çözülmesi gerekmeyebilir.

Viskoz, sıkıştırılamaz, serbest yüzey etkileri olmayan Newton tipi akışkanın daimi, laminer akışı için hareket denklemleri;

Süreklilik denklemi,

$$\nabla . V = 0 \tag{3.1}$$

Başka bir ifadeyle,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{3.2}$$

Navier-Stokes denklemi,

$$V . \nabla \ \ V = -\frac{1}{\rho} \nabla P' + \ \nu \nabla^2 V \tag{3.3}$$

Kartezyen koordinat sisteminde,

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = \ -\frac{1}{\rho} \frac{\partial P'}{\partial x} + u \ \ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \tag{3.4}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial P'}{\partial y} + v\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \quad (3.5)$$

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial P'}{\partial z} + v\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \quad (3.6)$$

Bu durumda denklem 3.2, 3.4, 3.5 ve 3.6 HAD ile çözülmesi gereken dört hareket denklemidir ve dört bilinmeyen u, v, w ve P' vardır. Eğer akış sıkıştırılabilir olursa bu denklemler daha farklı ifade edilir. Bunlara ek olarak enerji denkleminin de çözüme dâhil edilmesi gereken durumlar olmaktadır.

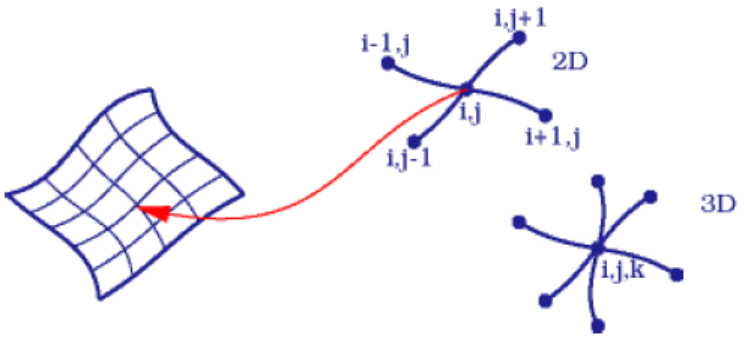
3.2 HAD Ayrıklaştırma Yöntemleri

Hesaplamalı akışkanlar dinamiği analizlerinde ayrıklaştırma için birçok yöntem arasında sonlu elemanlar, sonlu farklar ve sonlu hacimler yöntemi en sık kullanılan yöntemler arasındadır.

3.2.1 Sonlu Farklar Yöntemi

Diferansiyel denklemlerin çözümü için kullanılan ilk sayısal yöntemlerdendir. Modelin içinde barındırdığı sonsuz sayıdaki nokta sonlu sayıdaki nokta ile tanımlanarak temel korunum denklemlerinin diferansiyel formları noktalar üzerine uygulanabilir. Her noktanın yapısal olarak birbiriyle uygun şekilde bağlı olması gerekir. Hız, basınç gibi özellikleri taşıyan noktalar komşu noktaya bu özellikleri aktarmaktadır. Modelin sonlu sayıda noktaya ayrılarak çözüm aranmasından dolayı hassasiyet doğrudan çözüm alanının ayrıklaştırılmasıyla ilgilidir.

Sonlu farklar yöntemi basit ve hassas bir yöntemdir. Ancak yapısal bir çözüm ağına ihtiyaç duyduyğu için karmaşık geometrilere uygunlanması zordur.



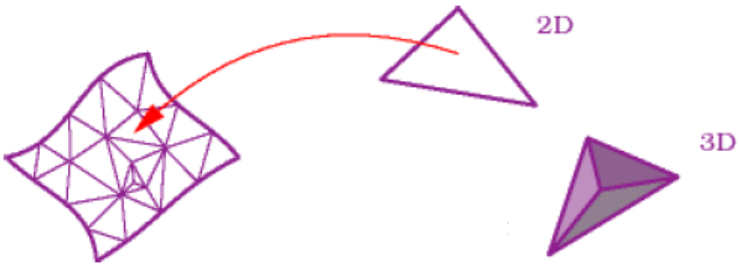
Şekil 3. 2 Sonlu farklar yöntemi [15]

3.2.2 Sonlu Elemanlar Yöntemi

Sonlu Elemanlar yöntemi temelde yapısal analizler için geliştirilmiş çok yaygın ve etkin bir sayısal yöntemdir. İlk olarak gerilme analizi problemine uyarlanmış olan yöntem matematikçilerden ziyade daha çok mühendislik uygulamalarında kullanılmaktadır. Gerilme analizinde deplasman alanı veya gerilme alanı; ısı analizinde sıcaklık alanı veya ısı akısı; akışkan problemlerinde ise akım fonksiyonu veya hız potansiyel fonksiyonudur.

Yöntemin uygulanabilmesi için model, davranışı belirlenmiş olan birçok elemana bölünür. Elemanların şekli çözüm alanının boyutuna, şekline, sınır koşulların özelliğine, problemi tanımlayan ana denklemlerin karakteristikleri dikkate alınarak seçilir.

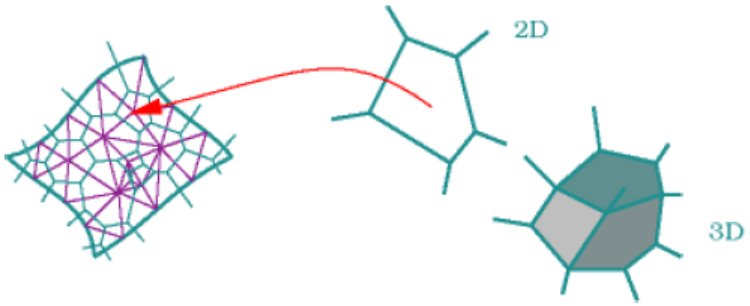
Sonlu elemanlar yönteminin altyapısı sonlu hacimler yöntemine göre daha karmaşık ve zahmetli matematiksel işlemler gerektirdiğinden daha çok uğraş ve donanıma ihtiyaç vardır. Sonlu elemanlar yönteminin sonlu farklar yönteminden farkı ise, bunun sınır değer problemini değil, varyasyonel problemi temel almasıdır.



Şekil 3. 3 Sonlu elemanlar yöntemi [15]

3.2.3 Sonlu Hacimler Yöntemi

Sonlu hacimler yöntemi kısmi diferansiyel denklemlerin cebri eşitliklere dönüştürülmesi ve çözümlemesi için kullanılan metotlardan biridir. Akış modeli ayrıklaştırılırken yapısal elemanlara ihtiyaç duymaması yöntemin avantajlarındandır. Sonlu elemanlar yönteminin de bu yönünün olduğunu biliyoruz ancak sonlu hacimler yöntemi uniform olmayan sayısal çözüm ağlarında daha iyi sonuçlar vermektedir. Sonlu hacimler yönteminde temel korunum denklemlerinin doğrudan ayrıklaştırılması prensibine dayandığından modellemenin iyi olmadığı durumlarda da iyi sonuçlar alınmaktadır. Bu yöntem çok sık kullanılan bir HAD ayrıklaştırma yöntemidir.



Şekil 3. 4 Sonlu hacimler yöntemi [15]

3.3 HAD Çözüm Adımları

Genel olarak HAD analizi çözümleri aşağıda belirtilen adımlardan oluşur:

1. Hesaplanmak istenen bölge belirlenir ve bu bölge üzerinde ağ örme işlemi yapılır. Ağ örme işlemi sonucu eğer bir alanda çalışılıyorsa sonlu sayıda küçük alan, eğer bir hacimde çalışılıyorsa sonlu sayıda küçük hacim elde edilir. Bu sonlu sayıdaki hacimler kontrol hacimlerini temsil eder. Her bir kontrol hacmine korunum denklemleri uygulanarak çözüm yaptırılır.
2. Oluşturulan geometri üzerindeki her bir kenar veya yüzeyde sınır şartları tanımlanır.
3. Akışkan tipi ve akışkan özellikleri belirlenir. Birçok HAD yazılımı veritabanında akışkan türü ve özelliklerini barındırmaktadır.

4. Sayısal değerler ve çözüm yolları belirlenir. Akış türüne ve karakterine göre değişiklik gösterecek olan çözüm algoritmalarının doğru seçilmesi gerekmektedir.

5. İterasyon işlemi için doğruluğu önemli olmayan başlangıç şartları tanımlanması gerekir. Bu başlangıç şartlarına göre yukarıda bahsedilen denklemler her bir kontrol hacminde döngüsel olarak çözülür. Eğer Navier-Stokes denkleminin terimleri tek tarafa toplanırsa, artık (residual) olarak adlandırılan bu terimlerin toplamı her bir kontrol hacmi için sıfır olduğunda, çözüm tam olur. Ancak bir HAD çözümünde çözüm hiçbir zaman sıfır olamaz, iterasyon sayısının artırılmasıyla sıfıra yaklaşılacağı umurlur. Transport denklemi çözümünün sıfırdan sapması artık (residual) olarak tanımlanabilir. Artıklar sıfıra yaklaştıkça çözümün doğruya yaklaştığı söylenebilir.

6. Hız ve basınç gibi akış alanı parametreleri çözüm yakınsadıktan sonra grafiksel olarak veya yazılım tarafından görsel araçlar kullanılarak çizdirilip analiz edilebilir. Bunun yanında yazılımın kullanıcıya çeşitli formülasyonlar ile sunacağı bazı özel fonksiyonlar da izlenebilir.

3.3.1 Ağ Oluşturulması

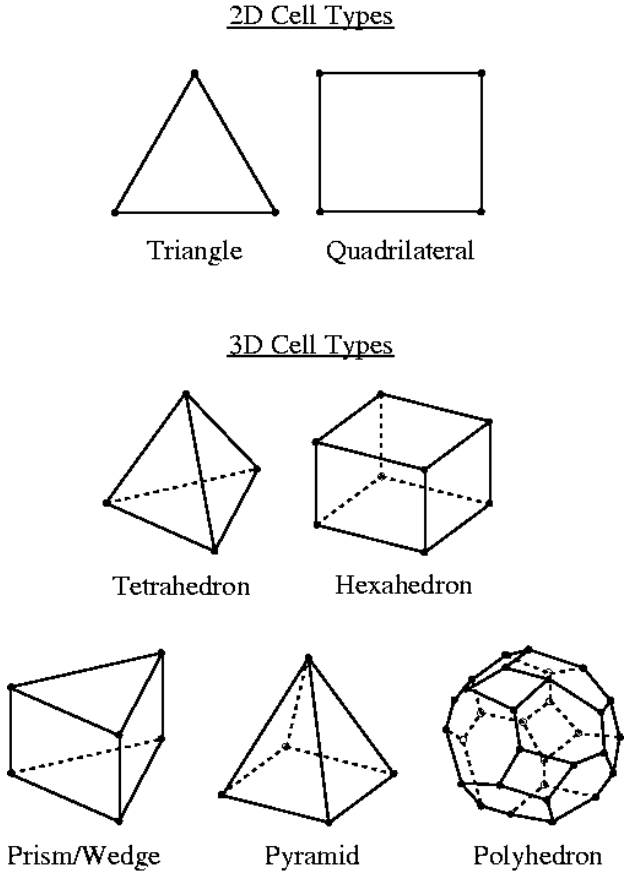
Hepsalama yapılacak bölgede kontrol hacimleri (hücreler) oluşturularak akış değişkenlerinin hesaplanması için ağ oluşturma işlemi yapılır. Ağ oluşturma işlemini HAD yazılımları kendi bünyesinde barındırabildiği gibi ayrı bir program vasıtasıyla da yapılabilir.

HAD analizinde elde edeceğimiz sonuçların yaklaşım doğruluğu, ağ içinde kullanılan eleman tiplerine (Şekil 3.5) ve sayılarına bağlıdır. Quadrilateral ve hexahedral şeklindeki elemanlar (aynı serbestlik derecesine sahip olduklarında) üçgen ve tetrahedral şekilli elemanlara göre genellikle daha iyi sonuç verir. Ayrıca hexahedral şekilli elemanların kullanılması eleman sayısını önemli miktarda düşürebilir. Bu düşüş sonucunda da analiz zamanı ve analiz sonrası işlem zamanı düşer. Bunlara ek olarak lineer-olmayan sistemler için ve elemanların dizilişinin fiziksel olarak önemli olduğu

yerlerde hexahedral ve quadrilateral şekilli elemanların kullanımı daha uygun olmaktadır.

Ağ oluştururken yakınsamanın sağlanabilmesi (artıkların sıfıra daha yakın değerlerde sağlanabilmesi) ve analizden doğru sonuç alınabilmesi için aşağıdaki hususların göz önünde bulundurulmasında fayda vardır.

- Oluşturulan ağ düzgün (uniform) bir şekilde dağılmış olmalıdır.
- Eğrisel yüzeylerdeki eleman tiplerinin seçiminde özellikle dikkat edilmesi gerekmektedir.
- Yapının durumuna göre simetriden yararlanılması faydalı olacaktır.
- Kullanılan elemanların büyüklüğünün hesaplama alanı içindeki değişiklikleri yansıtacak büyüklükte olması gerekmektedir. Az ya da fazla olması fazladan sayısal hesap yapılmasına, gereksiz zaman kaybına veya hesaplamadaki hatanın fazla olmasına sebep olabilir.

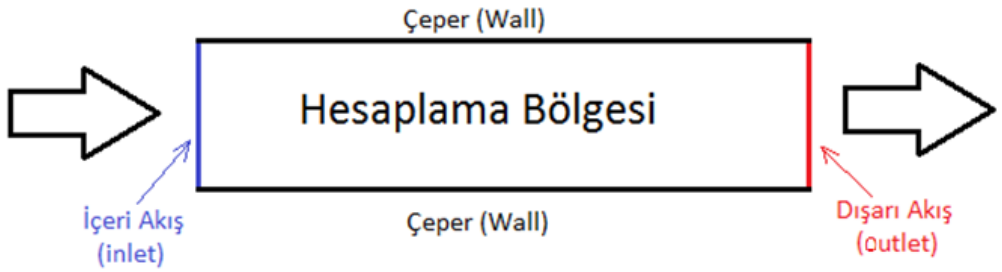


Şekil 3. 5 Ağ oluşturmada kullanılan eleman tipleri [16]

3.3.2 Sınır Şartları

İki HAD çözümü için, hareket denklemleri, hesaplama bölgesi ve hatta ağ bile aynı olabilmesine karşın, modellenen akışın türü uygulanan sınır şartları tarafından tayin edilir. Doğru bir HAD çözümü elde etmek için uygun sınır şartları gerekir [13].

Çoğu hesaplama bölgesinde en az bir defa da olsa kullanılan, basit bir sınır şartı olan çeper (wall) sınır şartı akışkan geçişinin olmadığı yüzeylerdir. Çeper sınır şartının tanımlandığı yüzey boyunca çepere göre hızın normal bileşeni ve teğetsel hız bileşeni sıfır alınır. İçeri akış (inlet) ve dışarı akış (outlet) sınır şartları akışkanın hesaplama bölgesine girdiği ve çıktığı yüzeylerdir. Bu bölgelerde genellikle hız veya basınç değişkenleri belirtilmesi gerekir. Bunlara ek olarak periyodik, simetrik, arayüz, iç sınır şartı gibi sınır şartları da vardır.



Şekil 3. 6 Basit bir hesaplama bölgesi için sınır şartları

3.3.3 Çözüm Modelinin Belirlenmesi

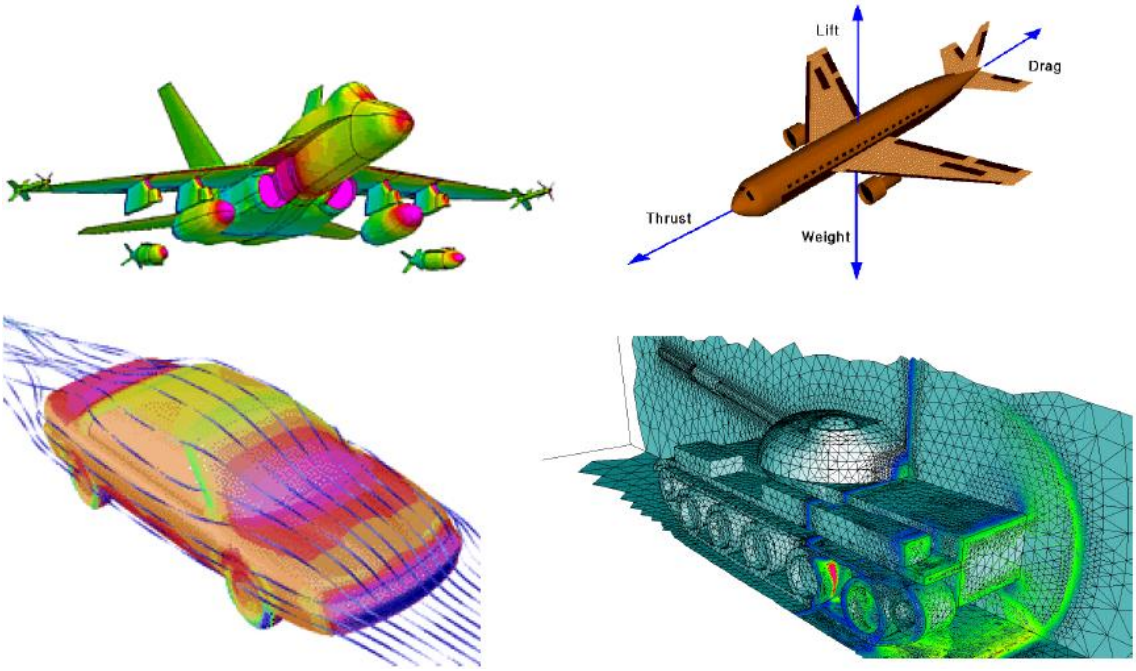
Laminer akışlar için HAD analizi mükemmel sonuçlar verir. Fakat oluşturulan ağ iyi çözünürlükte olmalı ve sınır şartları doğru tanımlanmalıdır. Laminer akışlar için tanımlı model çoğu HAD yazılımında doğrudan seçilebilir.

Türbülanslı akışların HAD analizinde türbülanslı akışlar için geliştirilmiş türbülans modelleri kullanılması gerekmektedir. Genel bir türbülans modeli maalesef yoktur. Türbülans modelinin seçimi birçok parametreye bağlıdır. Akışın karakteri, istenilen hassasiyet, kullanılacak donanımın kapasitesi, analiz süresi gibi kısıtlara bağlı olarak standartlaştırılmış türbülans modelleri arasından seçim yapılır. Standart türbülans modelleri uygulamadaki birçok mühendislik problemlerinde makul çözümler verir.

Türbülans modelleri hakkında çokça bilgiye gerek yazılım dokümanlarından gerekse yapılan birçok araştırma sonucu yayınlanan makalelerden kolayca ulaşılabilir. Kısaca bazı modeller: Doğrudan sayısal simülasyon (direct numerical simulation - DNS), büyük girdap simülasyonu (large eddy simulation - LES) ve navier-stokes denklemlerinden türetilen türbülans modelleri k-ε modeli, k-ω modeli ve q-ω modeli olarak sıralanabilir.

3.3.4 Son işlem (Postprocess) ve Doğrulama

Analizin bu adımı akış vektörleri, akış çizgileri, basınç gradyenleri, yüzeydeki sıcaklık, basınç, hız değişken dağılımları gibi pek çok datanın çıkarılmasını kapsar. Akışın simüle edilmesi, grafiksel olarak verilerin çıkarılması vb. işlemler yapılır.



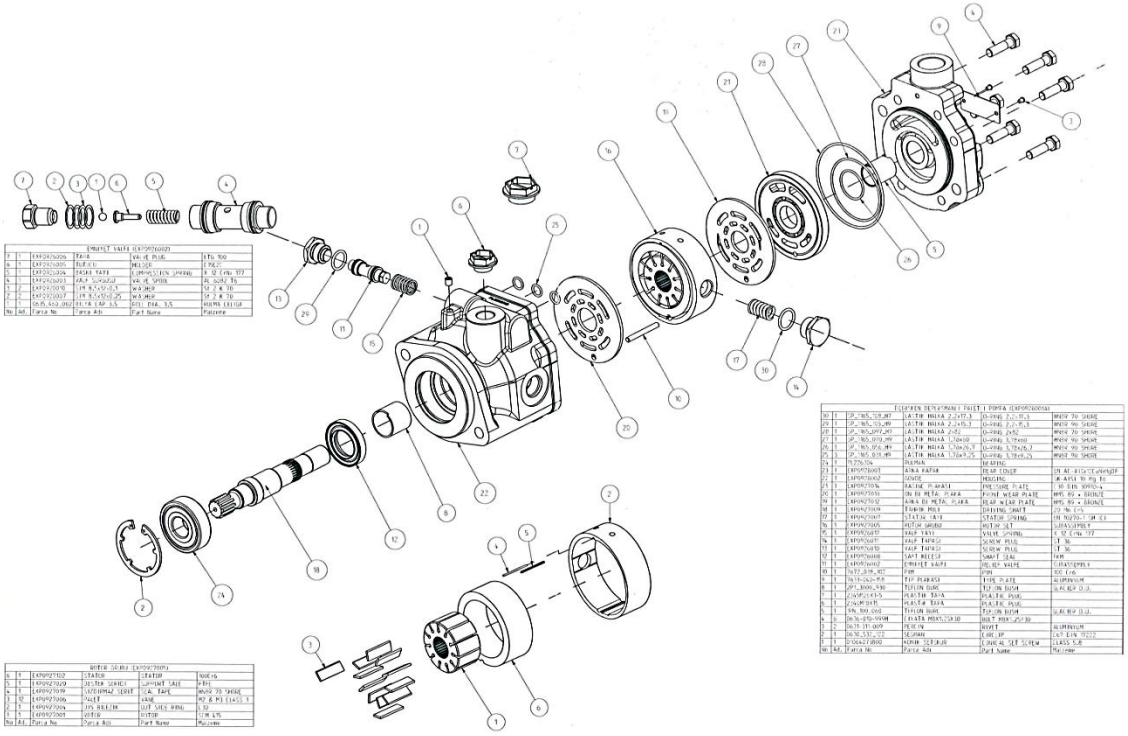
Şekil 3. 7 Çeşitli HAD analizi uygulamaları[15]

BÖLÜM 4

PALETLİ POMPA HAD ANALİZİ

Genelde bir pompa tasarımında debi, devir sayısı, çalışma basıncı gibi parametreler girdi olarak ele alınır. Bu parametrelerden yola çıkılıp çoğu zaman ampirik olarak diğer parametreler belirlenir. Bu da tasarımcı için yeterli olmamakta, spesifik durumlardan uzak kalınıp genel yaklaşımlar ele alınarak pompa tasarlanmaya çalışılmaktadır. Pompa tasarımında karşılaşılan zorlukların başında pompa içindeki akışın bilinmemesi gelmektedir. Hesaplamalı akışkanlar dinamiği ile desteklenen pompa tasarımlarında pompa içindeki akış ile ilgili özellikler tasarımcıya sunulmaktadır. HAD analizi, tasarım süreçlerinin kısaltılmasına oldukça fayda sağlanmaktadır.

Bu bölümde, tasarım sürecindeki Şekil 4.1 ’de patlatılmış resmi verilen paletli pompanın, hesaplamalı akışkanlar dinamiği analizi uygulaması ile kanat sayısı, dönme hızı gibi değişkenlerin akış üzerindeki etkileri incelenmektedir. Bölüm 3 ’te HAD analizi hakkında özet bilgi verilerek, HAD hakkında bilgi sahibi olunması amaçlanmıştı. Bu çalışma kapsamında yapılan analiz Bölüm 3 ’te değinilmeyen kullanıcı tanımlı fonksiyonların (user defined function - UDF) kullanımını içermektedir. Analizler Ansys Fluent 14.5 programı kullanılarak yapılmış olup, zamana bağlı (transient) olarak çözülmüştür.

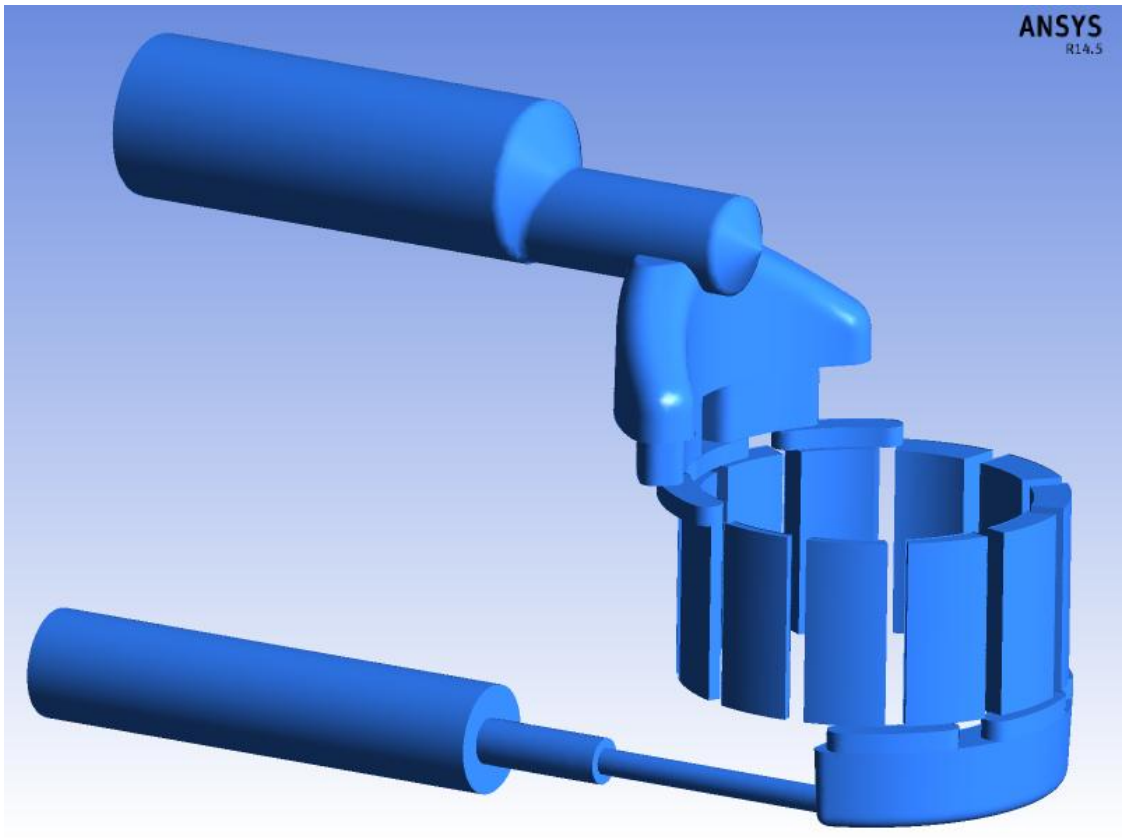


Şekil 4. 1 Paletli pompa patlatılmış resmi

Analiz adımlarını anlatmaya başlamadan önce bazı bilgilerin verilmesi faydalı olacaktır. İlk olarak çalışma kapsamında birden fazla analiz yapıldığını belirtmek gerekir. Şöyle ki, tasarlanan pompa 5, 8, 11, 14 ve 17 kanatlı olmak üzere 5 farklı şekilde çizilerek aynı analiz ayrı ayrı uygulanmıştır. Sonuçta basınç, hız, debi gibi değişkenlerin farklı kanat sayılarındaki değerleri incelenmiştir. Ayrıca pompa devir sayısının etkileri incelenmek üzere 1000 d/d, 1500 d/d, 2000 d/d, 2500 d/d, 3000 d/d, 3500 d/d, 4000 d/d ve 4500 d/d ’da analizler yapılmıştır.

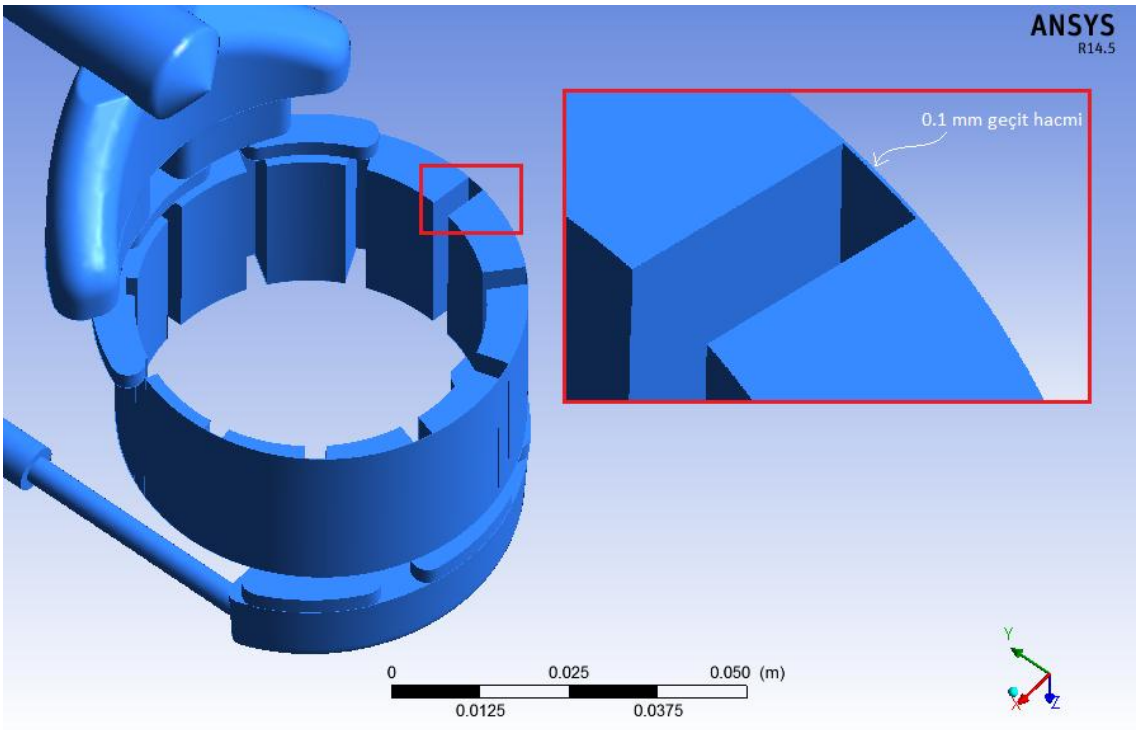
4.1 Geometrinin Oluşturulması ve Ağ (Mesh) Örme

Bilindiği üzere geometrinin oluşturulması için günümüzde kullanılabilen birçok CAD programı mevcuttur. Paletli pompanın geometrik çizimi yapıldıktan sonra akışkanın pompa içinde dolduracağı hacmin geometrisi alınarak Ansys Workbench 14.5 yazılımı vasıtasıyla ağ örme işlemleri yapılmıştır. Ön işlem olarak bilinen bu safha oldukça zaman alan, aynı zamanda analizin doğruluğunu önemli ölçüde etkileyecek işlemleri barındırmaktadır.



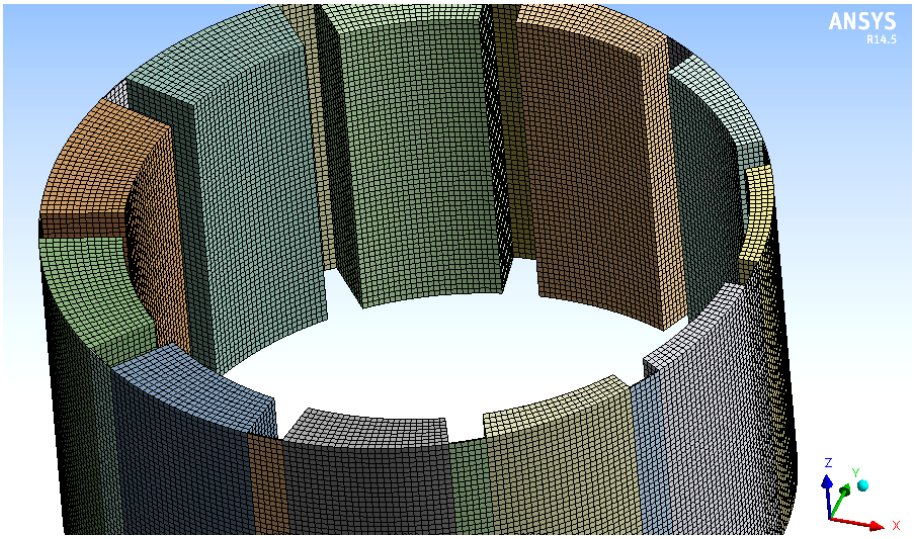
Şekil 4. 2 Akışkanın pompa içinde doldurduğu hacim

Hesaplama bölgesinin (pompa içinde akışkanın dolduracağı hacim) oluşturulması sırasında iki bölgeden (zone) bahsetmek gerekir. Bunlardan birincisi giriş ve çıkış kanalları olup sabit zonelardır. Diğeri ise pompanın hareketli kısmı olan rotor-stator bölgesidir. Bunlara ek olarak kanat ve stator (veya aşınması halkası) arasında 0,1 mm 'lik geçit hacmi oluşturulmuştur (Şekil 4.3). Ayrıca hareketli yüzeyler ile sabit yüzeyler arasındaki etkileşimi tanımlamak için arayüzler tanımlanması gerekmektedir. Bunların yanı sıra ağ örme işleminden önce geometrideki bütün yüzeylerin sınır şartları göz önünde bulundurularak tanımlanması iyi olacaktır. Fluent programında çeper sınır şartı için *wall*, içeri akış için *inlet*, dışarı akış için *outlet*, arayüzler için *interface* isimli yüzey tanımlamaları otomatik olarak algılanmaktadır, farklı isimlerle tanımlama yapıldığı takdirde program içinde değişiklik yapılabilmektedir.



Şekil 4. 3 Kanat ile Stator arasında oluşturulan geçit

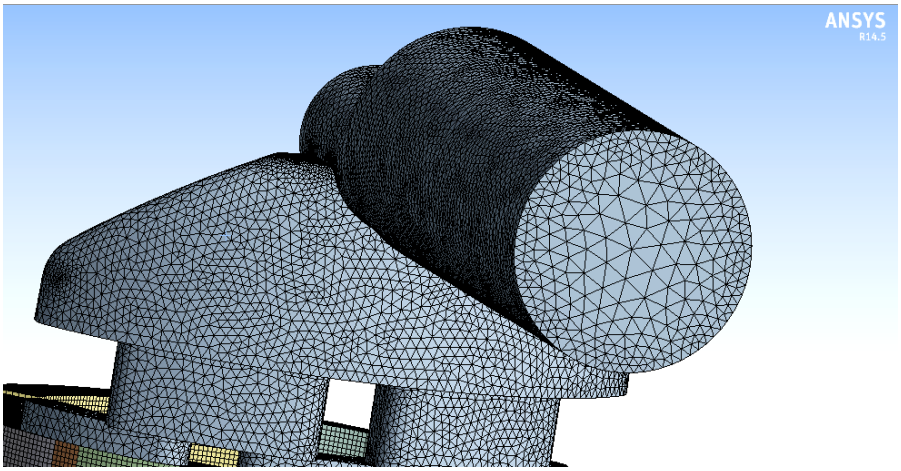
Ağ oluşturulurken rotor bölgesinde *mapped face meshing* uygulanmıştır (Şekil 4.4). Rotor-Stator bölgesinde ortalama 150.000 *hexahedral* hücresel eleman oluşturulmuştur. Eleman büyüklüğü 0,5 mm olarak belirlenmiştir.



Şekil 4. 4 Rotor-Stator bölgesi ağ örülmüş görünümü

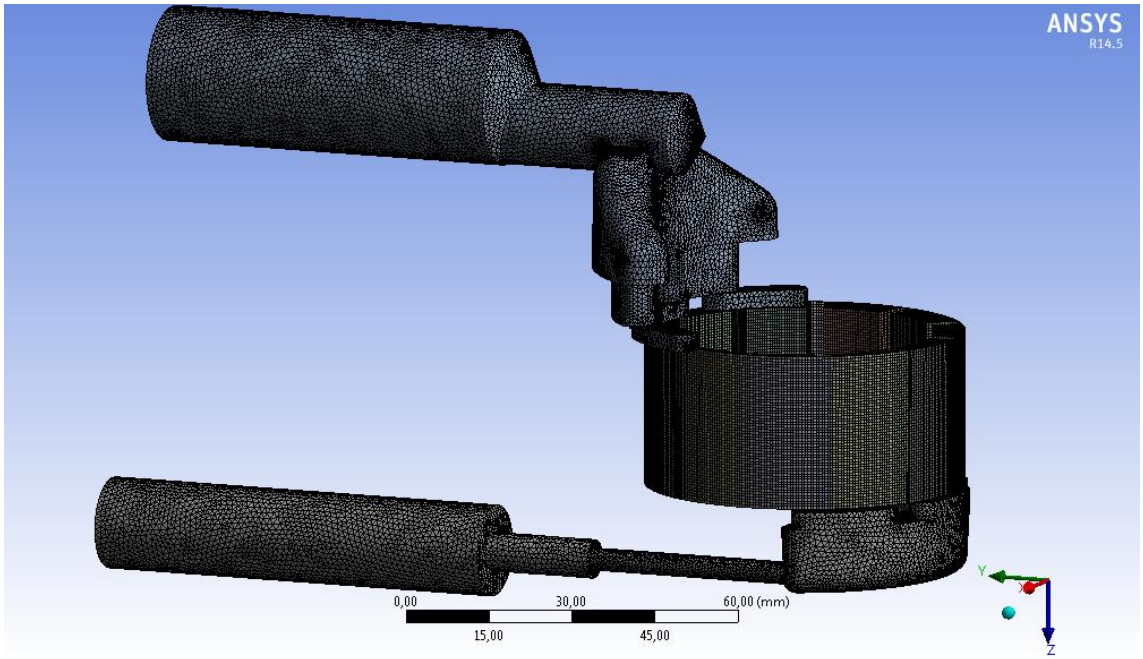
Pompa giriş ve çıkış kanallarında ise ortalama 450.000 *tetrahedral* hücresel eleman oluşturulmuştur (Şekil 4.5). Yüzeye yakın bölgelerde kayma gerilmeleri etkisi fazladır. Akışkan hareketinde yüzeye yakın bölgelerde hız değişim farkları daha fazla olmaktadır.

Bu sebeple giriş ve çıkış kanallarında yüzeye yakın bölgeler daha sık ağ ile örülmüştür. Bu bölgelerde eleman boyutu 1 mm olarak seçilmiştir.



Şekil 4. 5 Giriş kanalı ağ örülmüş görünümü

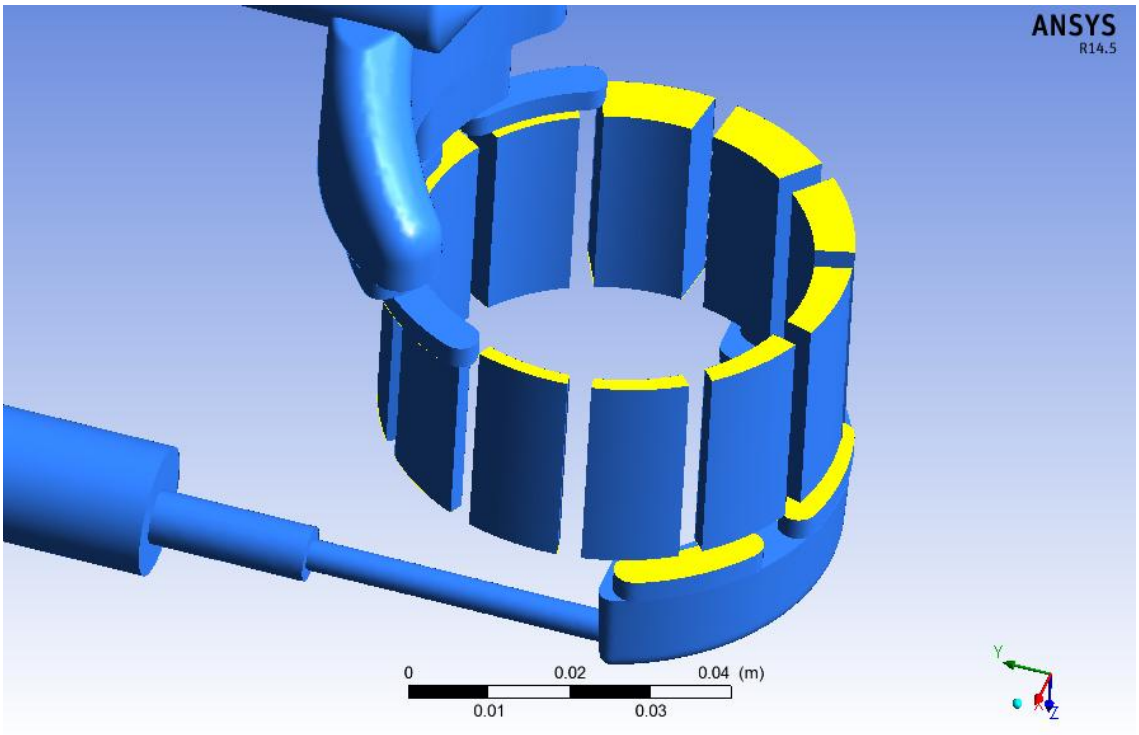
Analizi yapılan 5, 8, 11, 14 ve 17 kanatlı pompaların her biri ortalama 600.000 hücresel elemana sahiptir. Oluşturulan ağ yerine göre *quadrilateral*, *triangular*, *hexahedral* ve *tetrahedral* eleman tiplerini içermektedir (Şekil 4.6).



Şekil 4. 6 Ağ örülmüş paletli pompa

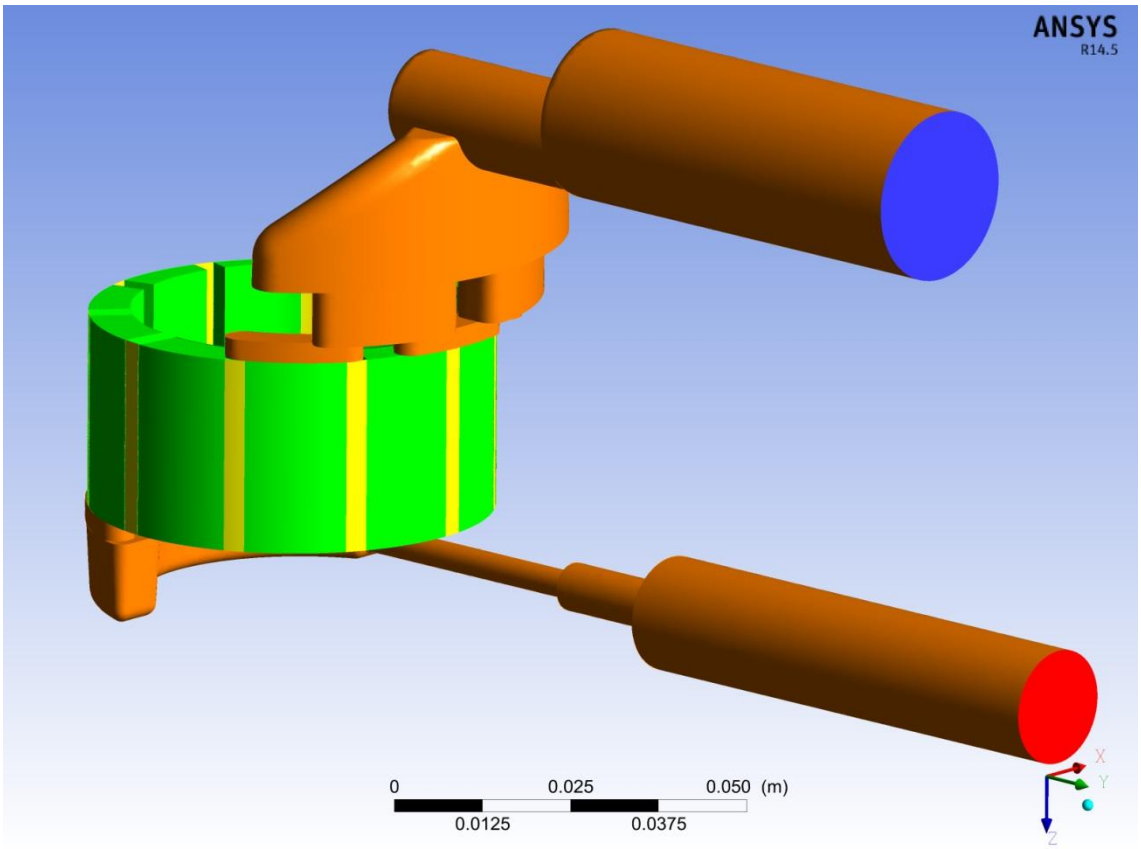
4.2 Sınır Şartları ve Akışkan Özellikleri

Birçok akışkan analizinde kullanılan çeper (wall), içeri akış (inlet), dışarı akış (outlet) sınır şartları tanımlanmıştır. Dışarı akış (outlet) sınır şartında akışkanın pompadan çıkış basıncı olan 185 bar girilmiştir. Pompa havuzu (kanat, rotor ve stator arasında kalan akışkan hacimleri pompa havuzu olarak adlandırılmıştır) ile giriş ve çıkış kanalları arasındaki yüzeyler arayüz (interface) olarak tanımlanmıştır (Şekil 4.7). Burada giriş-çıkış kanalları sabit, rotor hareketli olduğu için arayüz tanımlaması gerekmektedir.



Şekil 4. 7 Arayüz (interface) tanımlaması (sarı: arayüz)

Pompa içindeki bölgeler (zone) statik ve dinamik olarak ayrıldığı için ayrı ayrı tanımlanmıştır. *Interior-pump-cores, interior-pump-gaps* ve *interior-pump-pipes* olmak üzere toplam 3 adet hücresel zone oluşturulmuştur (Şekil 4.8). *Interior-pump-cores* ve *interior-pump-gaps* rotor hareketinden dolayı dinamik ağ bölgeleridir.



Şekil 4. 8 Hesaplama bölgesindeki bölge ve yüzey tanımlamaları
(mavi: içeri akış (inlet), kırmızı: dışarı akış (outlet), yeşil: rotor-stator bölgesi(interior-pump-cores), sarı: geçit hacmi(interior-pump-gaps), kahverengi: giriş-çıkış kanalları(interior-pump-pipes))

Pompa için seçilen akışkan daha çok otomotiv sektöründe kullanılan Shell Spirax S4 ATF HDX yağıdır, özellikleri Çizelge 4.1 'de belirtilmektedir.

Çizelge 4. 1 Akışkan özellikleri

Spirax S4 ATF HDX	
Yoğunluk [kg/m³]	847
Dinamik Viskozite [kg/m.s]	0.0134
Kinematik Viskozite [mm²/s] 40°C 100°C	 33.2 7.2

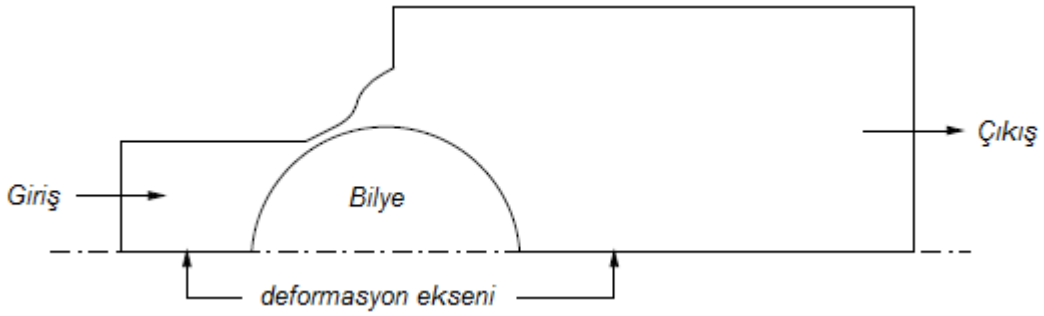
4.3 Kullanıcı Tanımlı Fonksiyon (User Defined Function - UDF)

Kullanıcı tanımlı fonksiyon, adından da anlaşılacağı üzere kullanıcının C programlama dilini kullanarak yazacağı bir programı çözüme dâhil etmesidir. Fluent, standart olarak yazılmış kodların dışında, kullanıcıya özel sınır şartları, kullanıcı tanımlı malzeme özellikleri, model parametrelerinin özelleştirilmesi, akış rejimiyle ilgili özel koşullar, son işlem bölümü için özel çıktılar gibi birçok değişkenin ve fonksiyonun hazırlanan program vasıtasıyla aktarılmasına olanak tanımaktadır. Bir veya birden fazla kaynak dosyası yüklenebilmektedir.

Doğada neredeyse bütün akışlar zamanla değişir. HAD analizlerinde zamandan bağımsız (steady) ve zamana bağlı (transient) olarak çözümler yapılmaktadır. *Steady* çözümler hesaplanması daha kolay, az zaman harcayan, güçlü donanım gerektirmeyen bir çözüm sunar. Daha güçlü bir analiz için *transient* analiz yapılması daha doğrudur. Bu çalışmada zamana bağlı (transient) çözüm yapılmıştır.

Yapılan analizlerde paletli pompa için C programlama dili kullanılarak özel tasarlanmış olan programlar ile dinamik mesh (ağ) uygulanmıştır. Rotorun hareketi UDF ile sağlanmıştır. Rotorun devir hızı ayarlanabilmekte olup, rotorun her 0.25 derecelik hareketinde çözüm yaptırılmıştır. Sonuçta, rotorun 0-90 derecelik hareketinde (360 adım) pompa içindeki yağ akışı analiz edilmiştir.

Dinamik mesh davranışlarının kontrolü, tanımlanmış 4 adet makro ile sağlanmaktadır. Tabiki bunlara ek başka fonksiyonların kullanımını gerektiren durumlar söz konusu olabilir. Basit bir örnek ile UDF uygulamasını anlamak adına: Şekil 4. 9 ’da gösterilen çekvalfin analizi için hazırlanan kodları inceleyebiliriz. 2 boyutlu olarak ele alınan modelde bilyenin hareketi UDF kullanılarak tanımlanmaktadır. Normalde akışkanın basıncı ile hareket edecek olan bilyeye UDF ile hareket kazandırılarak akışın davranışı simüle edilebilir. Burada iki çeşit fonksiyon kullanılabilir. Bilyenin hareketi için *DEFINE_CG_MOTION* ve eksen üzerinde bulunan nodların hareketini ve yeniden düzenlenmesini sağlamak için *DEFINE_GEOM* fonksiyonları kullanılabilir.



Şekil 4. 9 Problemin şeması [16]

UDF Kodları;

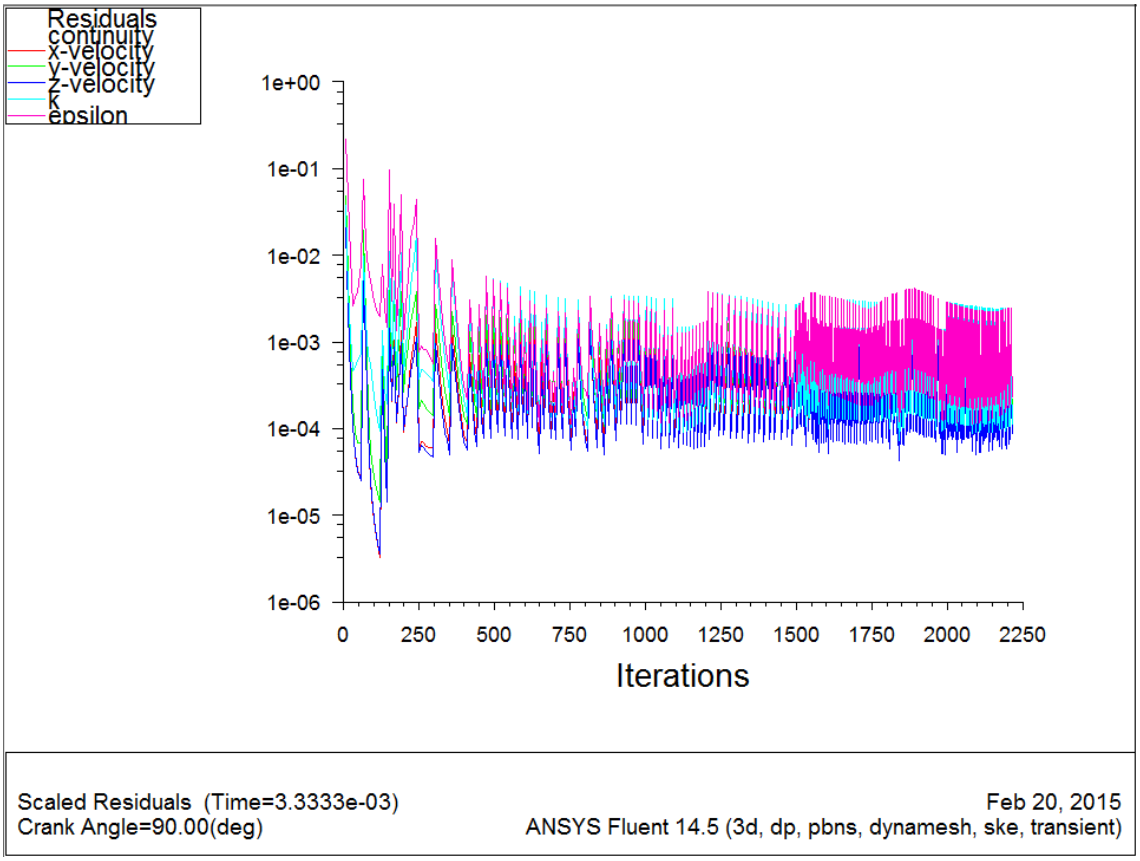
```
#include <udf.h>

DEFINE_CG_MOTION(valve, dt, cg_vel, cg_omega, time, dtime)
{
    real freq = 0.25;
    real amp = -0.075;

    real omega;
    real vel;

    omega= 2.0*M_PI*freq;
    vel = amp*sin(omega*time);

    cg_vel[0] = vel;
    cg_vel[1] = 0.0;
    cg_vel[2] = 0.0;

    NV_S (cg_omega, =, 0.0);
}

DEFINE_GEOM(axis1, domain, dt, position)
{

    position[1] = 0;
}

DEFINE_GEOM(axis2, domain, dt, position)
{

    position[1] = 0;
}
```

4.4 Çözüm Modeli ve Metodunun Belirlenmesi

İki denklemli bir model olan Standard k–ε Türbülans Modeli oldukça yaygın kullanılan bir model olup bu çalışmada da seçilen modeldir. İki denklemli denilmesinin nedeni kütle ve momentum (gerekiyorsa enerji) denklemlerinden başka iki adet daha

transport denkleminin çözümünü içermesindendir. Bu denklemler türbülans kinetik enerjisi *k* ve türbülans yitim hızı *ε* değişkenlerini getirmektedir.

Fluent ’te çözücü tipleri basınca dayalı veya yoğunluğa dayalı olmak üzere iki şekildedir. Genel olarak düşük hızlı sıkıştırılamaz akışlar için basınca dayanan, yüksek hızlı sıkıştırılabilir akışlar için yoğunluğa dayanan çözücü seçilir demek yanlış olmaz. Çalışmamızda basınca dayanan (pressure-based) çözücü seçilmiştir. Basınca dayalı çözücü için 5 adet algoritma (pressure-velocity coupling algorithms) mevcuttur: SIMPLE, SIMPLEC, PISO, Coupled, Fractional Step (FSM). Zamana bağlı çözümlerde en uygun algoritma PISO ’dur. Özellikle büyük zaman aralıkları seçilerek yapılacak çözümler için uygundur. Bunlara ek olarak basınç standart ve momentum, türbülans kinetik enerji, türbülans yitim hızı oranı ikinci derece ayrıklaştırma (second order upwind) yöntemi uygulanmıştır. Her adım (toplam 360 adım) için iterasyon sınırı 60 olarak ayarlanmıştır. Birçok analiz incelendiğinde ortalama 15-20 iterasyondan sonra çözümün yakınsadığı görülmektedir.



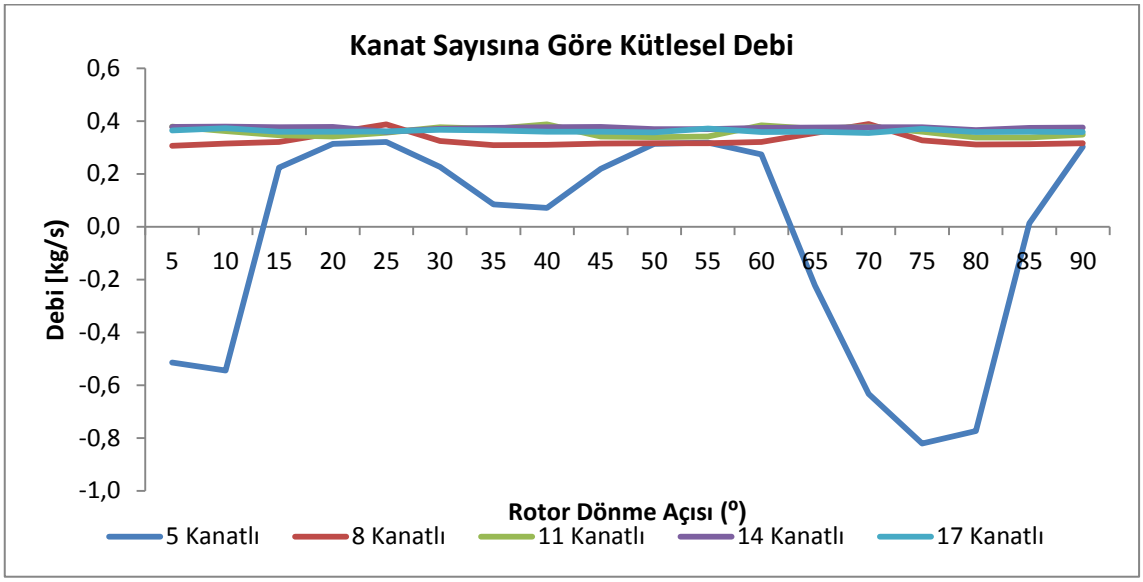
Şekil 4. 10 11 kanatlı paletli pompanın (4500 d/d) analizi - artıklar (residuals)

4.5 Analiz Sonuçları

Sonuçlar Fluent yazılımının kendi içerisinden alınabildiği gibi başka yazılımlar vasıtasıyla da alınabilir. Bu çalışmada Fluent ile birlikte Ansys CFD-Post kullanılmıştır.

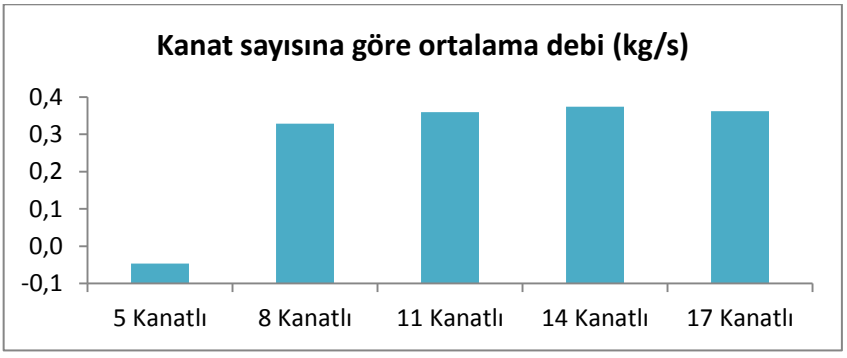
4.5.1 Kanat Sayısına Göre Sonuçların İncelenmesi

5, 8, 11, 14 ve 17 kanatlı paletli pompalar 3000 d/d dönme hızında bütün koşullar aynı olmak kaydıyla çözdürülmüştür. Debi ve pompa içi en yüksek basınç değişimleriyle ilgili veriler elde edilmiştir.

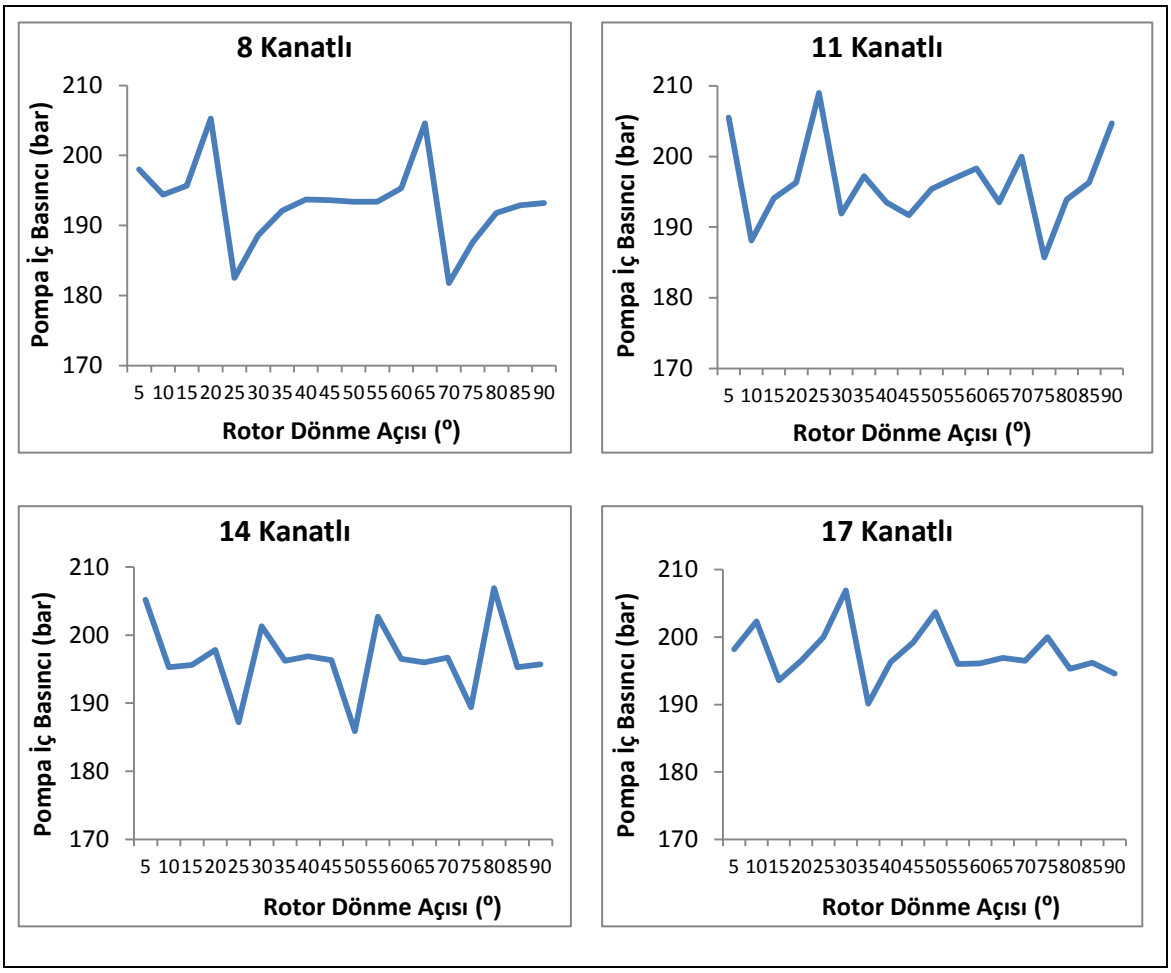


Şekil 4. 11 Kanat sayısına göre debi değişimi

5 kanatlı pompada akışın tersine döndüğü görülmektedir (Şekil 4.10). 8, 11, 14 ve 17 kanatlı pompalarda debi değerleri birbirine çok yakındır. Buradan da akışkan akımında her hangi bir olumsuzluk (akışın terse dönmesi gibi) gözlenmiyorsa kanat sayısıyla debinin çok fazla değişmediğini söyleyebiliriz (Şekil 4.11).



Şekil 4. 12 Kanat sayısına göre ortalama debi değişimi



Şekil 4. 13 Kanat sayısına göre pompa iç basınç değişimleri

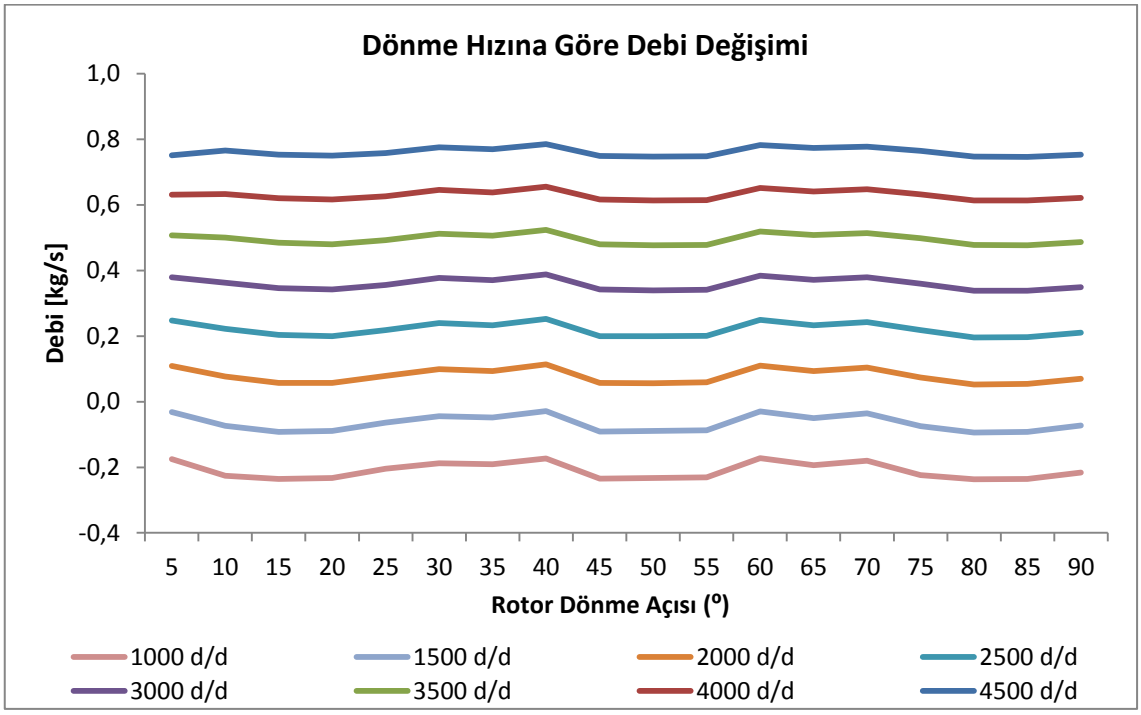
Pompa çıkışı 185 bar olarak tanımlanmıştı. Pompa içerisinde genelde 185 bar 'ın üzerinde bir basınç oluşmaktadır. Paletlerin pompa içindeki konumuna göre iç basınç sürekli değişmekte olup, belirli bir aralık içinde kalmaktadır. Kanatların giriş ve çıkış kanalı ağzına gelip geçmesiyle artan ve azalan bir eğri çizilmektedir. Kanat sayısının iç basıncı çok az miktarda etkilediğini söyleyebiliriz. Kanat sayısı arttıkça basınç değişim aralığının azaldığı görülmektedir (Şekil 4.12 ve Çizelge 4.2).

Çizelge 4. 2 Kanat sayısına göre pompa iç basınç değişimleri

Kanat Sayısı	**8 Kanatlı**	**11 Kanatlı**	**14 Kanatlı**	**17 Kanatlı**
Maksimum Basınç (bar)	205,3	209,0	206,9	206,9
Minimum Basınç (bar)	181,8	185,7	185,9	190,1
Fark Basıncı (bar)	23,5	23,3	21,0	16,8
Ortalama Basınç (bar)	193,2	196,2	196,5	197,7

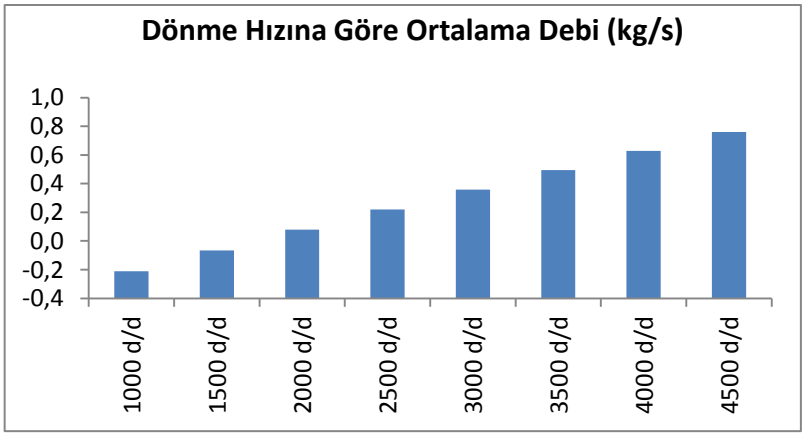
4.5.2 Dönme Hızına Göre Sonuçların İncelenmesi

11 kanatlı paletli pompa modelinin kullanıldığı analizler 1000 d/d, 1500 d/d, 2000 d/d, 2500 d/d, 3000 d/d, 3500 d/d, 4000 d/d, 4500 d/d olmak üzere 8 farklı dönme hızında çözdürülmüştür.

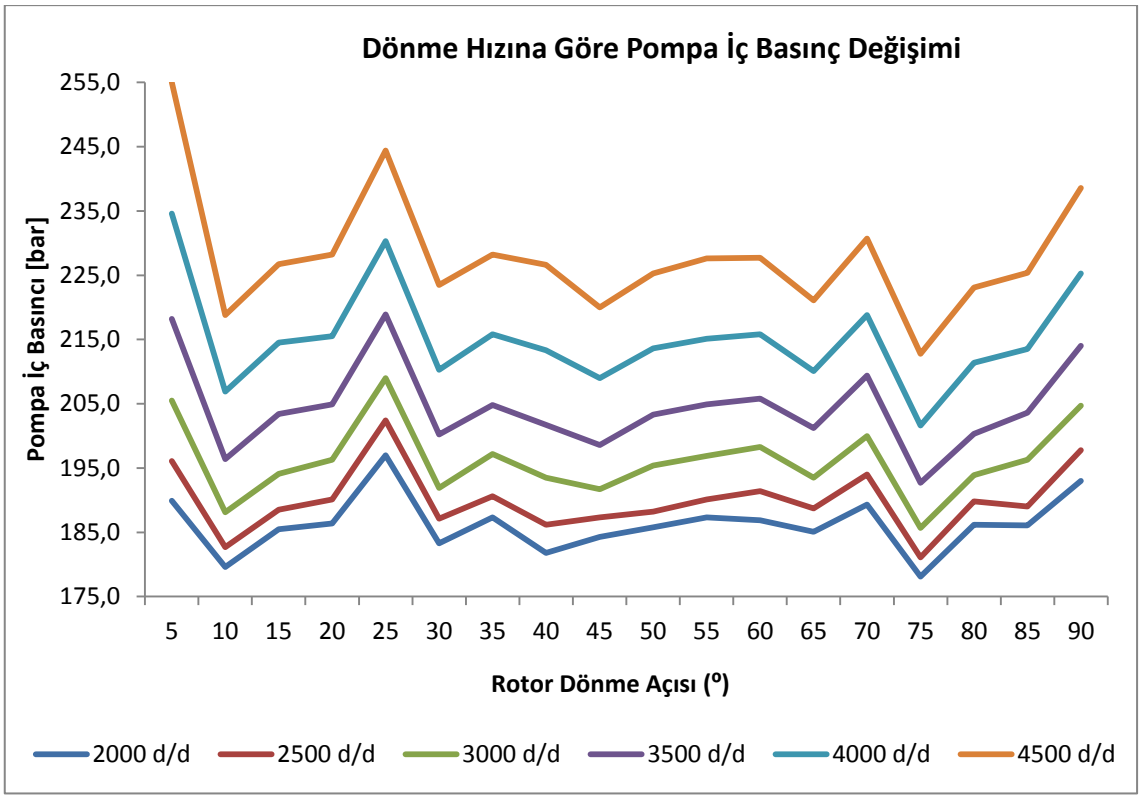


Şekil 4. 14 Dönme hızına göre debi değişimi

Beklenen şekilde dönme hızı arttıkça debi de artmaktadır. Ancak pompanın yağ basması için belirli bir hıza ulaşması şarttır. 11 kanatlı pompa analiz sonuçlarına göre çizdirilen Şekil 4.13 ve Şekil 4.14 'teki grafikler incelendiğinde yaklaşık olarak 1600 d/d 'dan sonra pompanın çalıştığını görüyoruz.



Şekil 4. 15 Dönme hızına göre ortalama debi değişimi



Şekil 4. 16 Dönme hızına göre ortalama pompa iç basınç değişimi

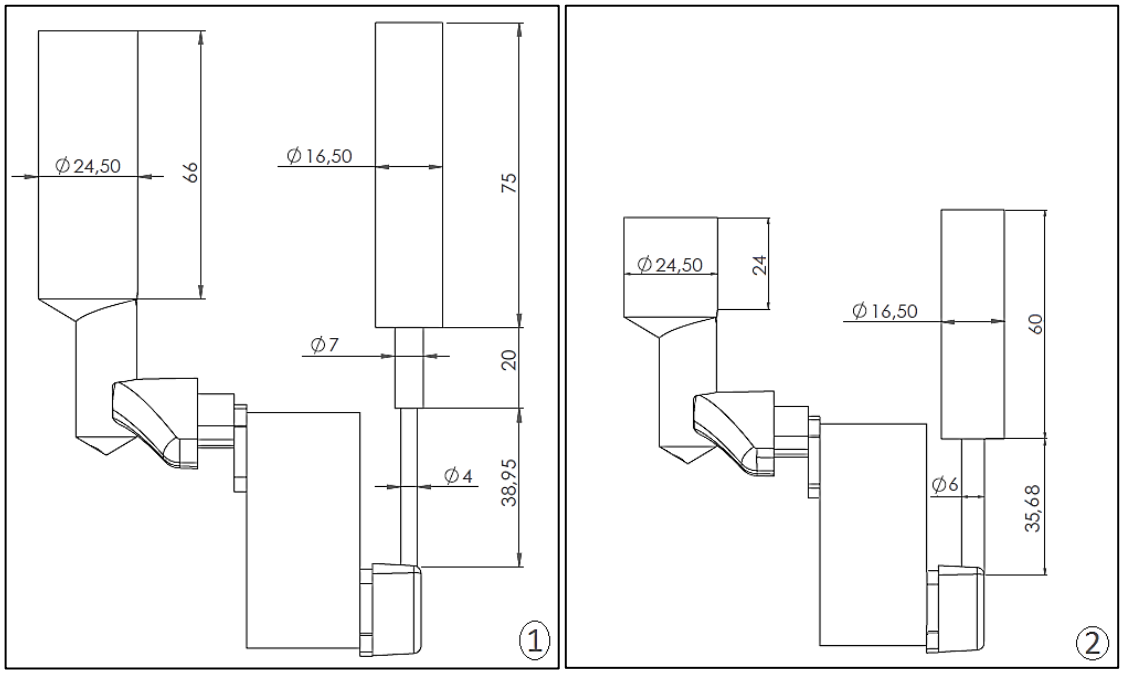
Pompa içindeki basınç dönme hızı arttıkça artmaktadır (Şekil 4.15). Dönme hızına bağlı olarak iç basıncın artarak artan bir eğri çizdiği görülmektedir. Basınç değişimleri kanatların konumlarıyla ilgili olup pompanın devri arttıkça iç basınç değişim aralığı da artmaktadır (Çizelge 4.3).

Çizelge 4. 3 Dönme hızına göre pompa iç basınç değişimleri

Dönme Hızı	**2000 d/d**	**2500 d/d**	**3000 d/d**	**3500 d/d**	**4000 d/d**	**4500 d/d**
Maksimum Basınç (bar)	197,0	202,4	209,0	218,9	234,6	255,0
Minimum Basınç (bar)	178,1	181,1	185,7	192,7	201,6	212,8
Fark Basıncı (bar)	18,9	21,3	23,3	26,2	33,0	42,2
Ortalama Basınç (bar)	186,3	190,1	196,2	204,6	215,3	228,0

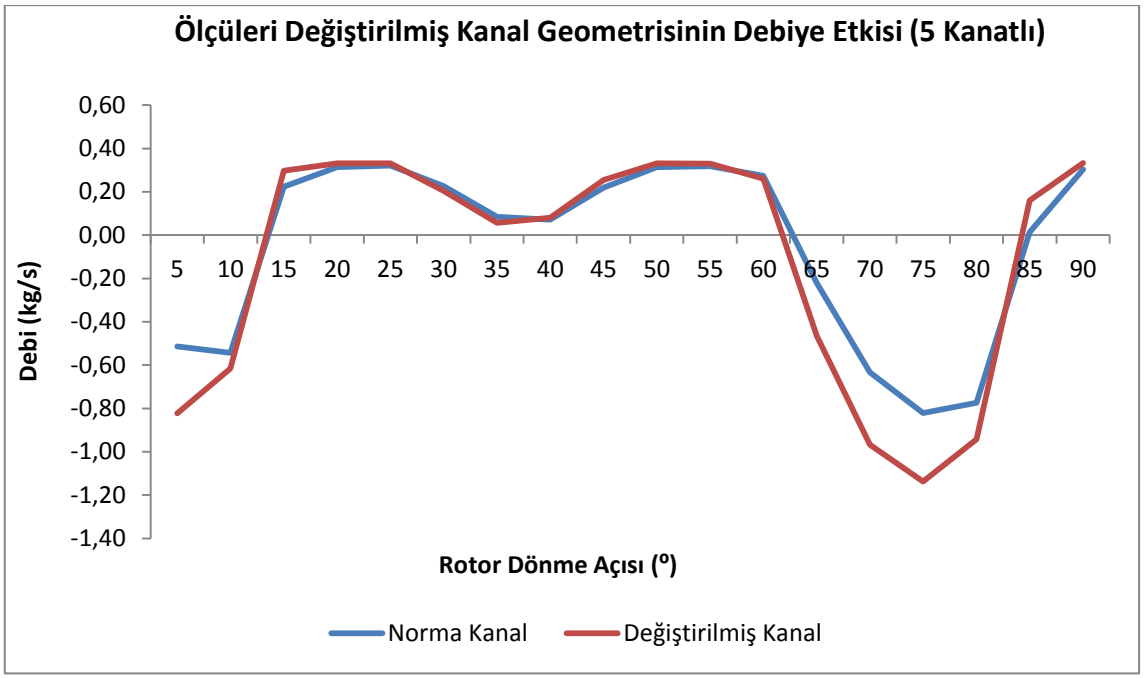
4.5.3 Pompa Çıkış Kanalı Boyutlarının Değiştirilmesinin Etkisi

5 kanatlı pompada akışın tersine dönmesini önlemek için çıkış kanalı geometrisinde bazı değişiklikler yapılmıştır. Çap artırılıp, kanalın boyu kısaltılmıştır (Şekil 4. 16). Ancak akış hareketinde ciddi bir değişiklik gözlenmemiştir.

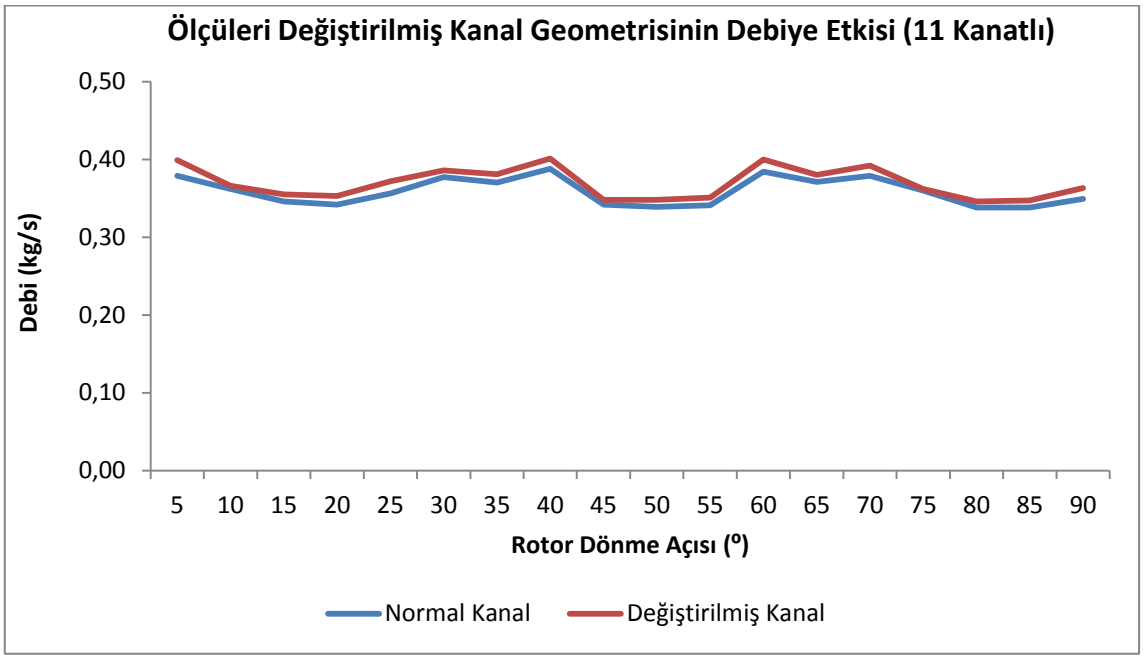


Şekil 4. 17 Pompa giriş ve çıkış kanalı ölçüleri (1- Normal, 2- Değiştirilmiş)

5 Kanatlı pompada Şekil 4. 17 'de görüleceği üzere rotor açısı 65 ile 85 derece arasındayken tersine akış olmaktadır. Kanal boyutlarının değiştirilmesi akımı istenenin aksi yönde etkilemiştir. Yani kanal geometrisinde yapılan değişiklik akışı çok fazla etkilememiş sonuçta pompada hâlâ tersine akış meydana gelmektedir.



Şekil 4. 18 Ölçüleri değiştirilmiş kanal geometrisinin debiye etkisi (5 kanatlı)



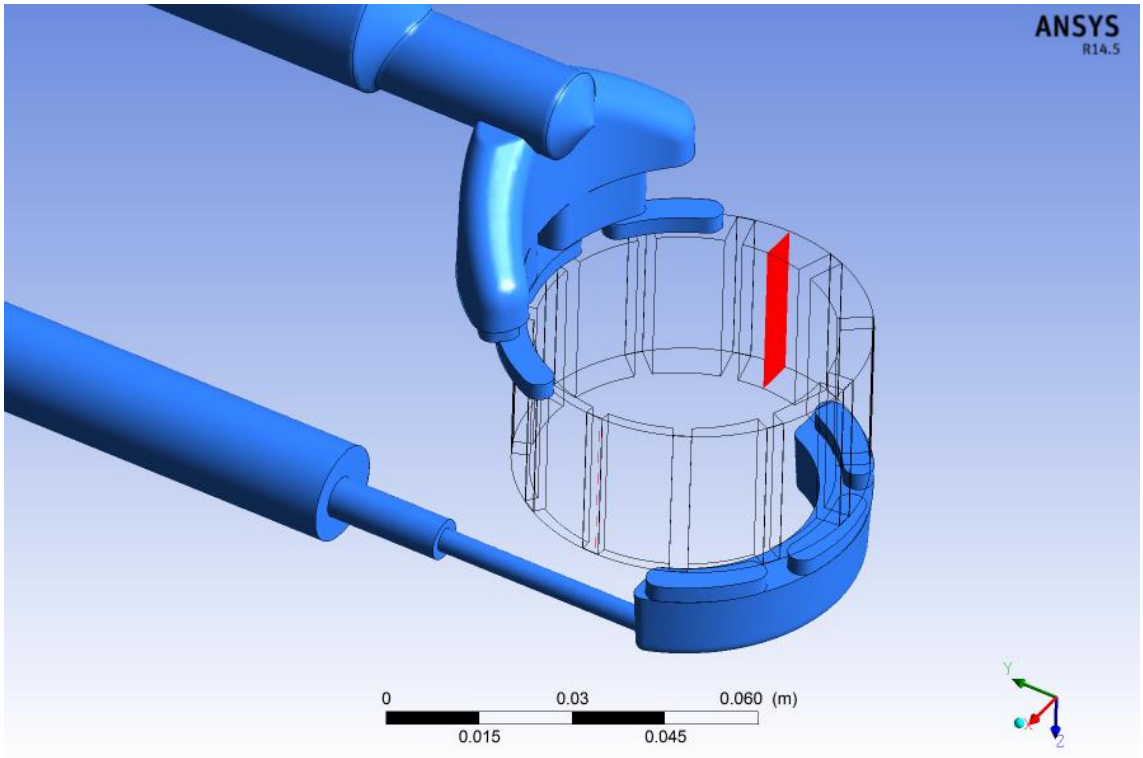
Şekil 4. 19 Ölçüleri değiştirilmiş kanal geometrisinin debiye etkisi (11 kanatlı)

11 Kanatlı pompada kanal ölçülerinin değiştirilmesinin debi değişimine etkisi çok az olmuştur (Şekil 4. 18). En dar kesitin çapı 2 mm artırılmasından dolayı debi çok az artmıştır.

4.5.4 Akışkan Akımı ile İlgili Sonuçlar

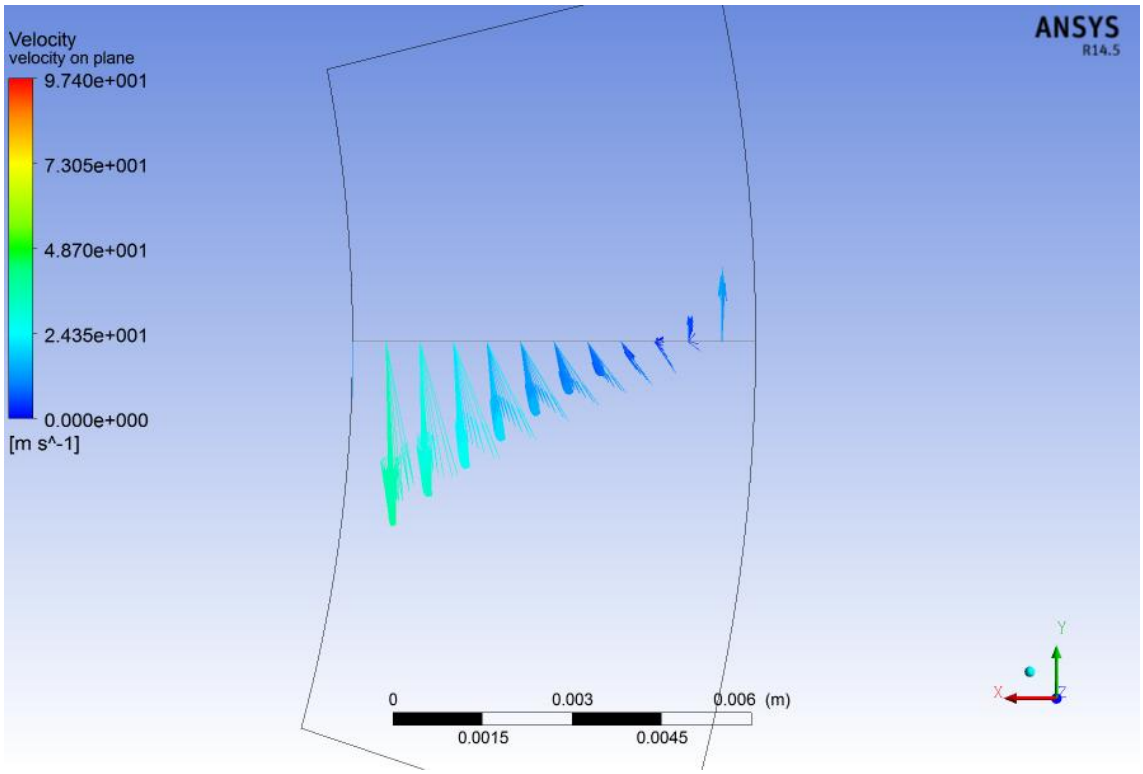
Debi, basınç ve hız değişkenleri kıyaslandığında en iyi değerlerin 11 kanatlı pompaya ait olduğunu söyleyebiliriz. Bu bölümde kanat sayısı 11 olan paletli pompa ile ilgili sonuçlar verilecek olup, diğer pompalarla ilgili sonuçlar ek olarak çalışmanın sonunda verilmiştir.

Dönme hızı 4500 d/d olan pompada Şekil 4. 19 'da kırmızı renk ile gösterilen yüzey tanımlanmıştır. Bu tanımlanan yüzey üzerinden hız profilleri alınmıştır. Bu işlem yapılırken pompa giriş ve çıkış kanalına bağlantısı bulunmayan bir havuz (iki kanat arası akışkan hacmi) dikkate alınmıştır.

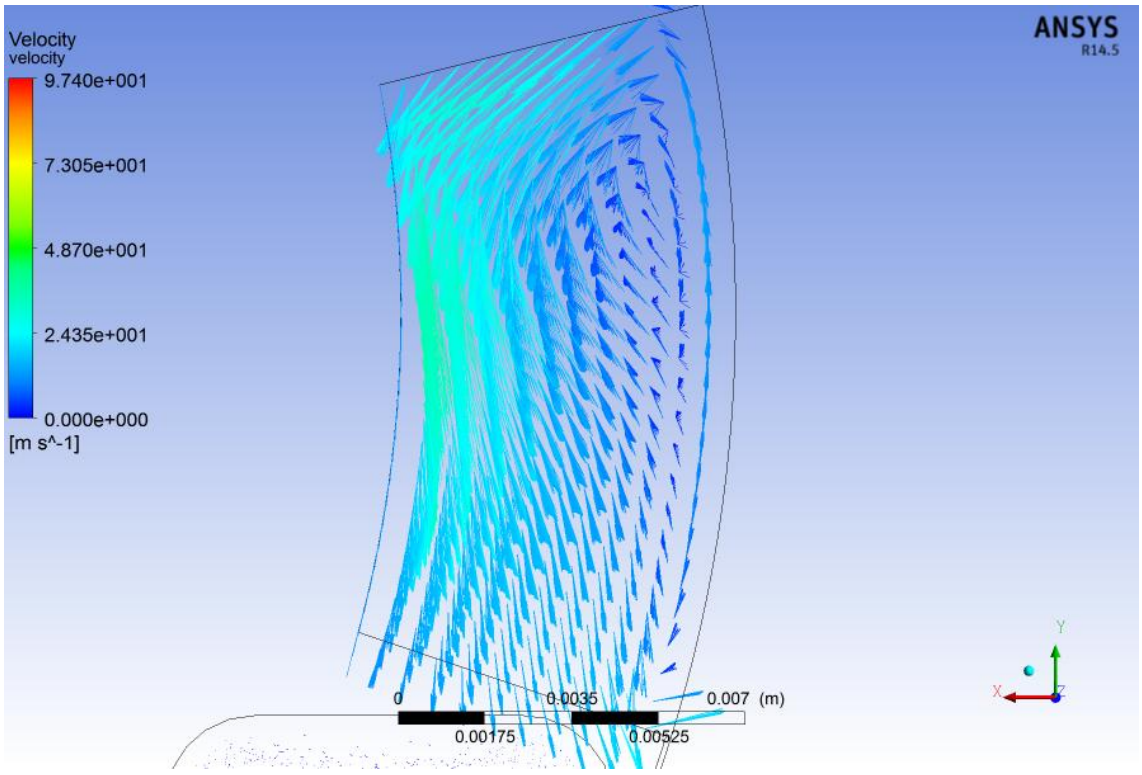


Şekil 4. 20 Hız profili alınan kesit

11 Kanatlı pomda içinde 11 adet havuz bulunduğuna göre rotorun hareketi ile pompa havuzları zaman zaman giriş ve çıkış ağzından uzaklaşmakta, akışkan bu havuzda hapsolmaktadır. Bu durumda havuz içerisindeki akışkan Şekil 4. 20 ve Şekil 4. 21 'de gösterilen hız profiline sahip olmaktadır. Görüleceği üzere akışkan havuz içinde karmaşık bir akış sergilemekte ve girdap oluşmaktadır. Burada rotor yüzeyi ve kanatlar hareketli, gövde yüzeyi de hareketsizdir. Bu yüzden hız vektörlerinin bu şekilde çıkması normaldir.

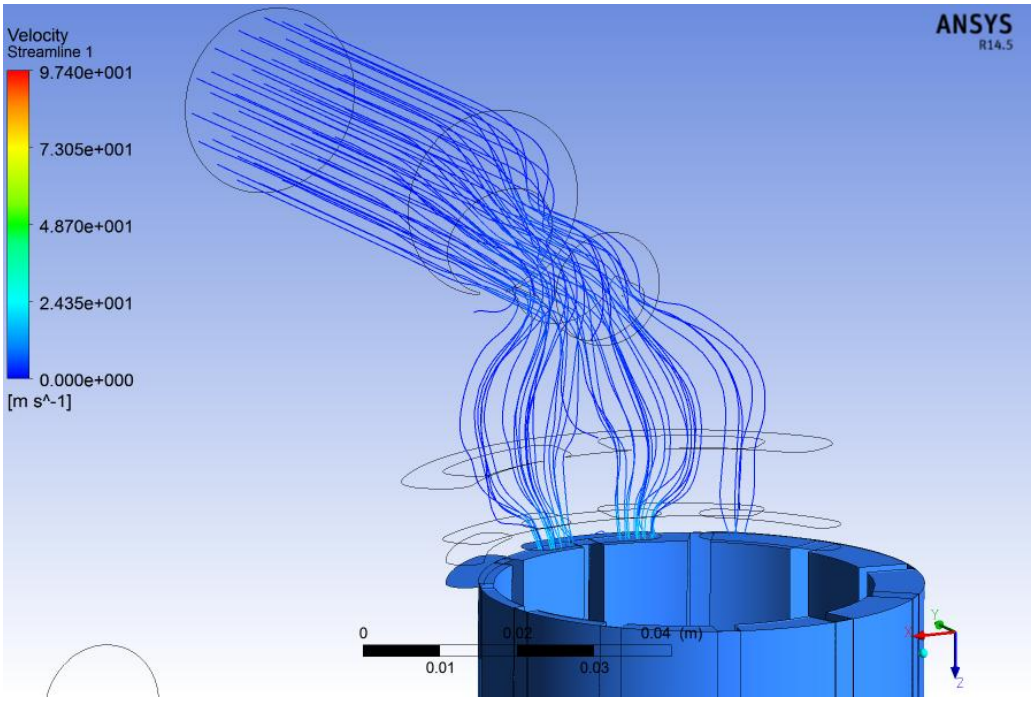


Şekil 4. 21 Pompa havuzu kesitindeki hız profili

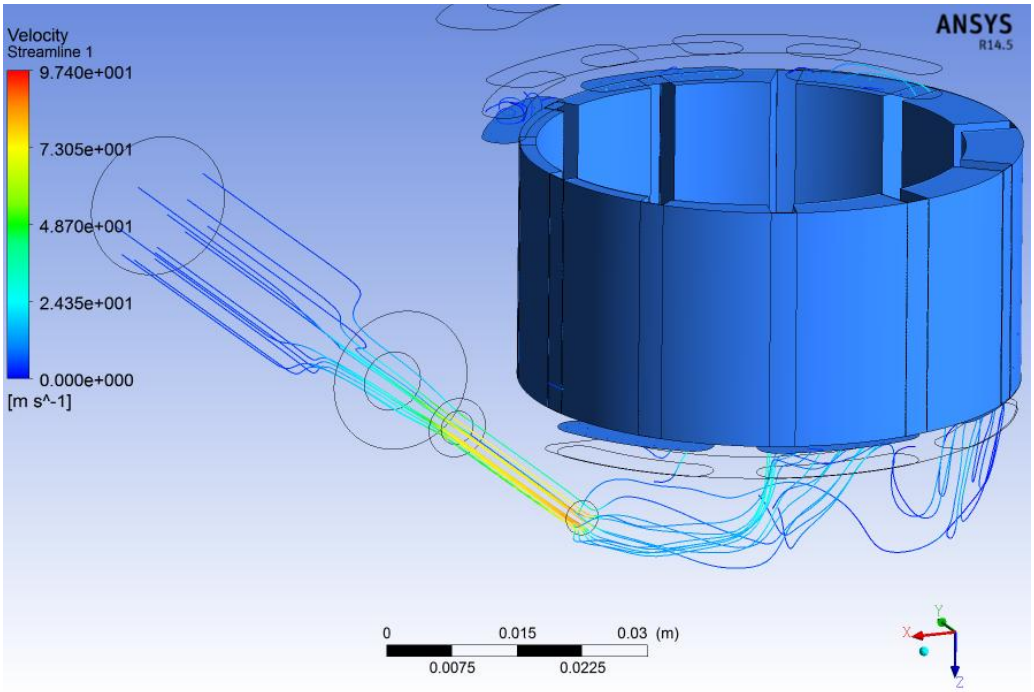


Şekil 4. 22 Pompa havuzundaki hız vektörleri

Şekil 4. 24 ’te akışa ait akım çizgileri gösterilmiştir. Rotorun farklı konumları için akışkan akımı giriş ve çıkış kanallarının rotor-stator bölgesine geçişinde kısmen farklılık gösterecektir ancak Şekil 4. 22 ve Şekil 4. 23 ’te gösterilenden çok da farklı olmamaktadır.

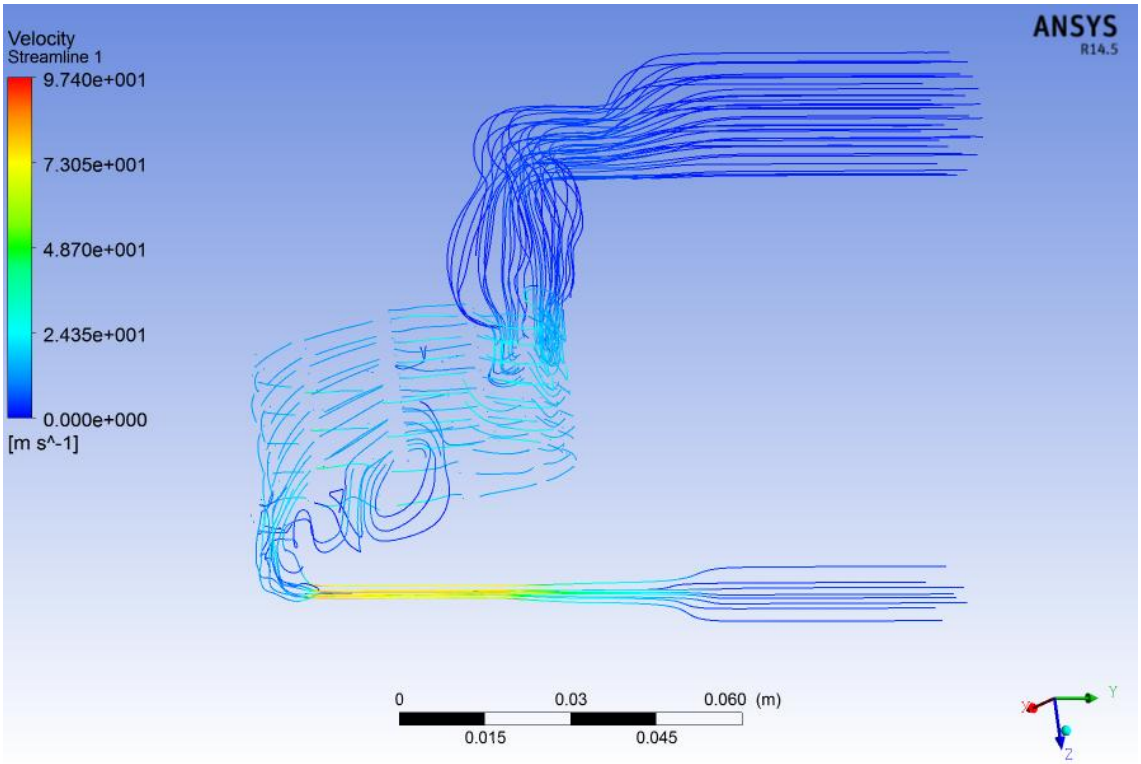


Şekil 4. 23 Giriş kanalı boyunca akım çizgileri



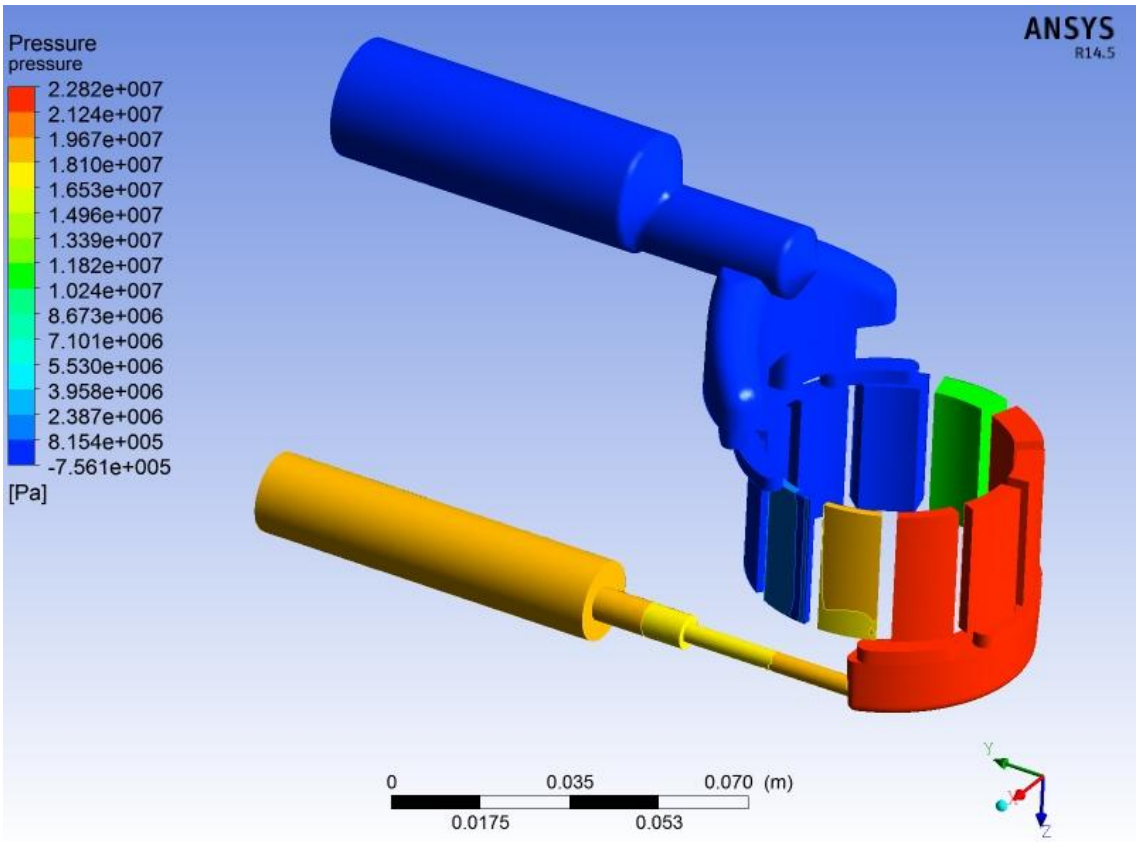
Şekil 4. 24 Çıkış kanalı boyunca akım çizgileri

Özellikle akışkan pompa havuzlarından geçtikten sonra çıkış kanalına akarken oldukça karmaşık bir akış sergilemektedir (Şekil 4. 23 ve Şekil 4. 24). Bunun başlıca sebebi akışkanın rotordan aldığı ivme ve sonrasında başka bir yola girmesi gösterilebilir. Bunun yanında tam bu bölgede basınç en yüksek değerine ulaşmaktadır (Şekil 4. 25).



Şekil 4. 25 Akışa ait akım çizgileri

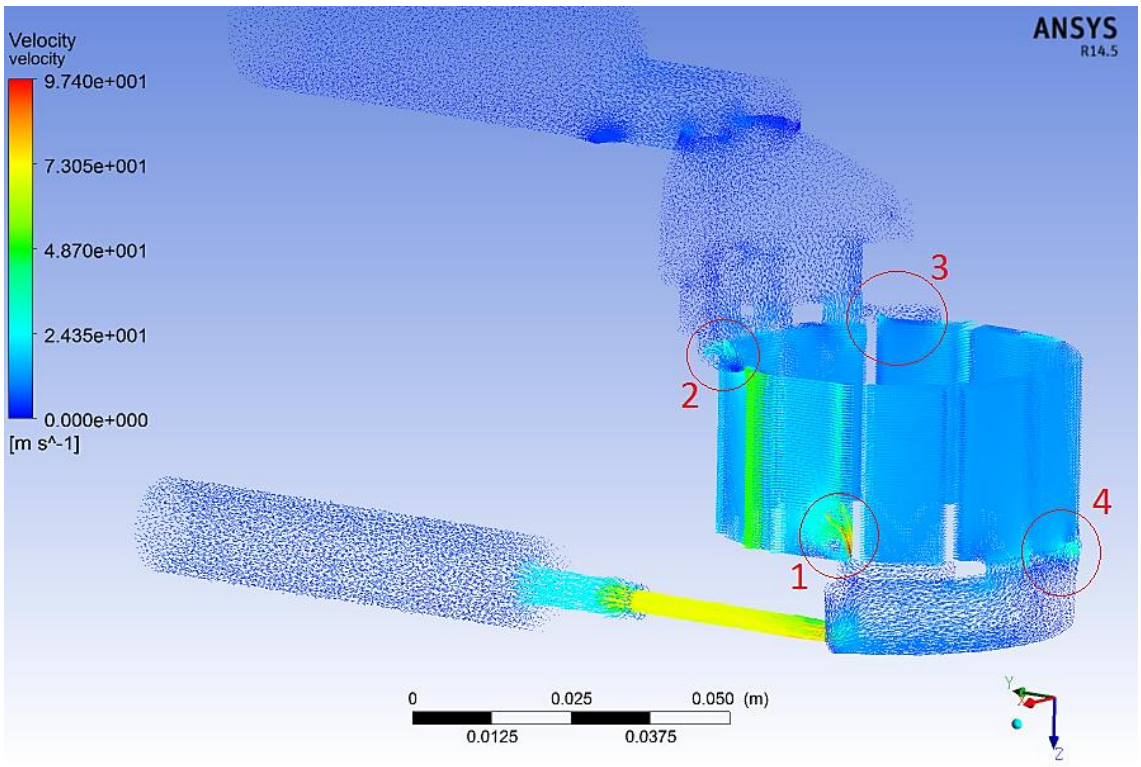
Pompa içerisinde en yüksek basıncın gözlendiği bölge statorun çıkış kanalına bağlandığı yerdir. Akışkan bu bölgeden geçtikten sonra ani basınç kayıpları oluştuğu takdirde pompada kavitasyon olayı meydana gelebilecektir. Nitekim analizlerimizde, kanatların çıkış kanalı ağzından ayrılırken ani basınç düşüşü olduğu görülmüştür (Şekil 4. 25).

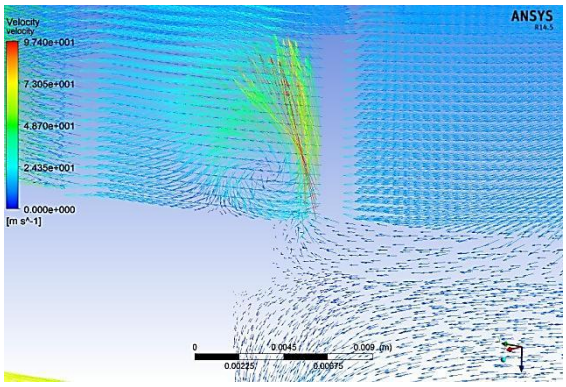


Şekil 4. 26 Pompa genelinde basınç dağılmı

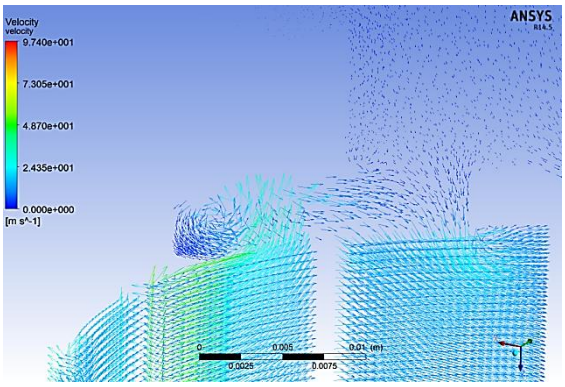
Akış ayrılmalarının görüldüğü, akışın türbülanslı bir yapı sergilediği bölgeler özellikle pompa giriş ve çıkış kanalları ile dönel hareketin meydana geldiği rotor-stator bölgesi arasındaki geçiş bölgeleridir. Bu bölgelerde tersine akışlar, girdap oluşumları gibi akışkan akımı olayları görülmektedir (Şekil 4. 26). Akışın en hızlı olduğu bölge ise kesitin dar olmasının etkisiyle çıkış kanalında bulunan en dar kesitli dairesel kanaldır.

Burada gösterilen sonuçlar dönme hızı en yüksek girilen analizlere aittir. Bu şekilde seçilmesinin nedeni sonuçların daha belirgin görülmesi içindir. Diğer hızlarda da söz konusu durumların birçoğu hıza bağlı olarak etkisi azalarak görülmektedir.

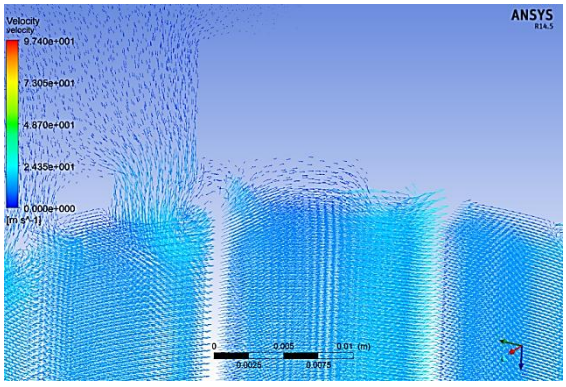




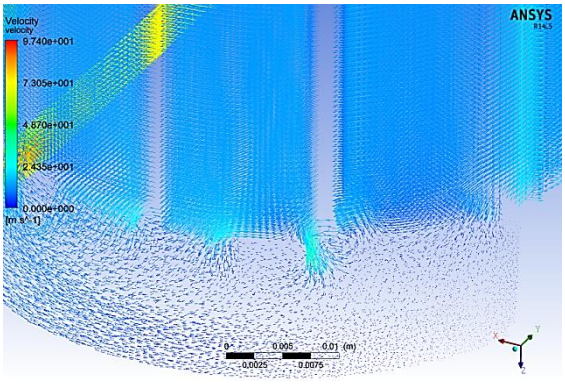
1



2



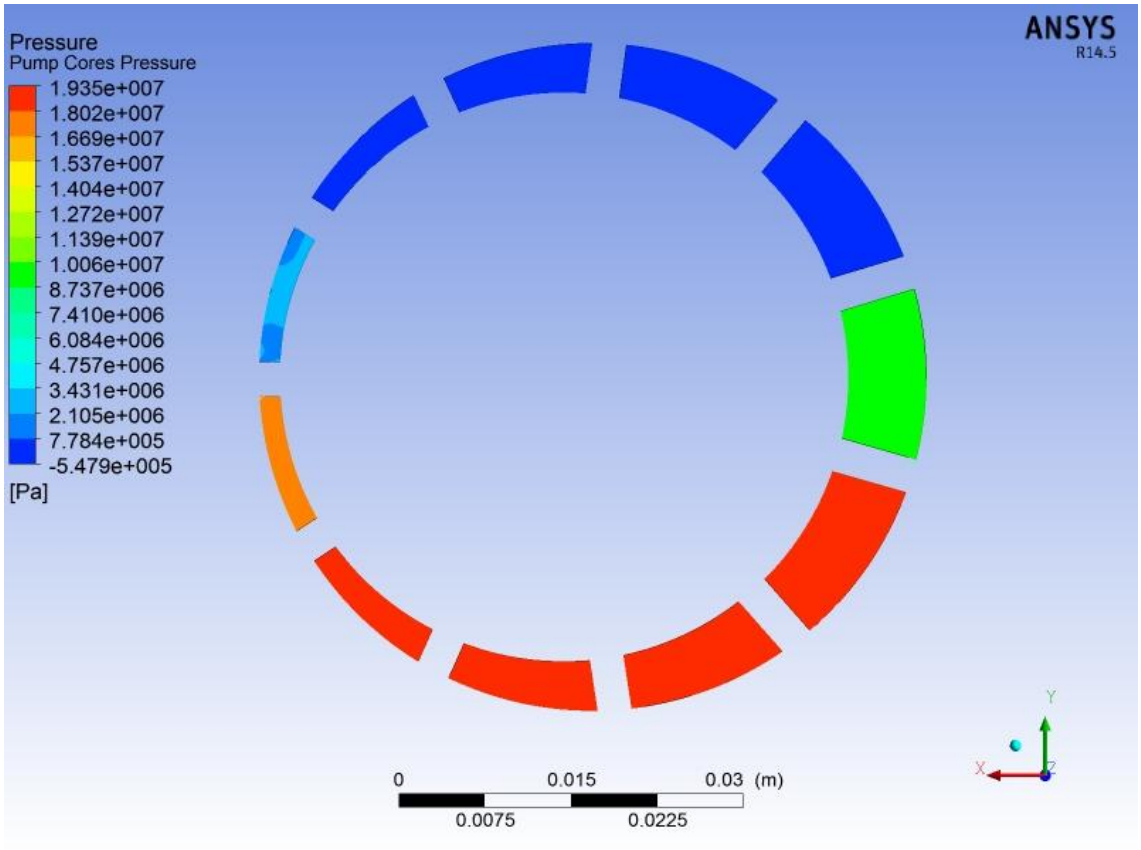
3



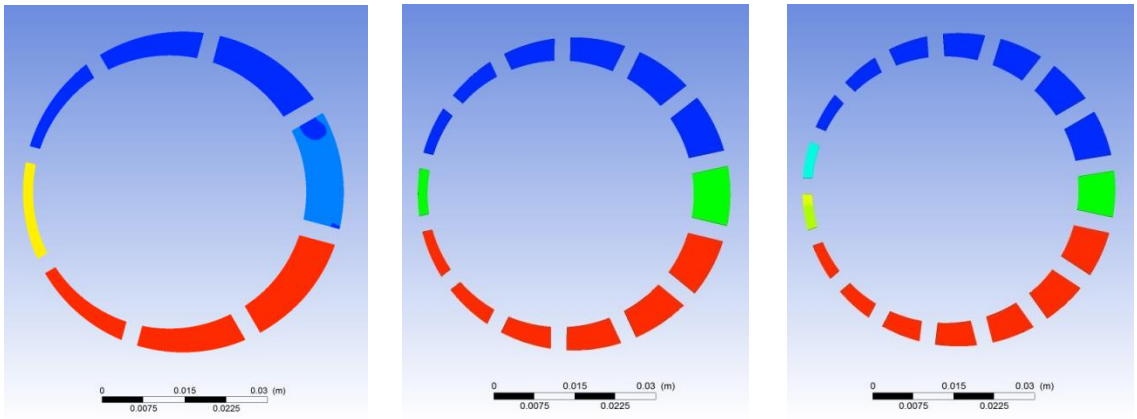
4

Şekil 4. 27 Akışa ait hız vektörleri

Şekil 4. 27 ’de rotor-stator bölgesindeki basınç dağılımları, farklı kanat sayıları için verilmiştir. Burada tüm modellerde benzer dağılımlar görülmektedir. Sonuçta giriş kanalına yakın bölgede basınç düşük, çıkış kanalına doğru gidildikçe ise basınç yükselmektedir.



11 Kanatlı



8 kanatlı 14 kanatlı 17 kanatlı

Şekil 4. 28 Kanat sayısına göre rotor-stator bölgesi basınç dağılımları

BÖLÜM 5

SONUÇ VE ÖNERİLER

Paletli pompa tasarıma yönelik yapılan bu çalışmada paletli pompalarda kanat sayısının pompa debisi ve pompa iç basıncına etkisi araştırılmıştır. 5, 8, 11, 14 ve 17 kanatlı olmak üzere 5 alternatif üzerinde çalışılmıştır. Sonuç olarak;

- Kanat sayısının az olmaması gerektiği ortaya çıkmıştır. Az kanat sayılı pompada akışın terse döndüğü görülmüştür.
- Kanat sayısı çoktan aza doğru gidildikçe akışın daha darbeli (kesikli) olarak iletildiği görülmektedir. Yani 8 kanatlı olan pompa 11 kanatlı olana göre daha kesintili bir akış sergilemektedir, basılan ortalama debi ise neredeyse değişmemektedir. Belirli bir kanat sayısına sahip pompalarda darbeli akış hemen hemen kaybolmaktadır. Bu çalışmada 11 kanatlı olan pompada darbeler hissedilmeyecek kadar az olup 14 ve 17 kanatlı pompayla hemen hemen aynıdır. Buradan sonuçla, kanat sayısını artırmanın pek faydalı olmayacağı, maliyetleri artıracağı aşikârdır.

Çalışmanın ikinci adımı da, dönme hızının debi ve pompa iç basıncına yönelik araştırmalarıdır. 11 kanatlı paletli pompa modeli ele alınarak farklı dönme hızlarında analizler yapılmıştır.

- Beklendiği gibi devir sayısının artmasıyla basılan debi ve pompa iç basıncında artış meydana gelmektedir.
- Pompanın emme yapabilmesi için devir sayısının belirli bir değere ulaşması gerekmektedir. Aksi durumda pompa emiş yapamamaktadır. Her pompa ve akışkan için minimum dönme hızı değişecektir.

- Pompa devrinin artmasıyla akışın darbeli halden normal hale doğru gittiği görülmektedir. Ancak darbeli akıştan bahsederken sistem hassasiyetinin göz önünde bulundurulması faydalı olacaktır. Birçok sistem için farkedilmeyecek ölçüde kesintili akış görülecektir.

Analiz sonuçlarından hareketle, giriş ve çıkış kanalı boyutlarının değiştirilmesi akışı çok fazla etkilememektedir. Kanal geometrilerinde yapılan küçük değişikliklerin pompa performansı üzerine etkisi çok az olmuştur. Pompa debisinde neredeyse hiçbir değişiklik olmamıştır. Fakat kanalların rotor-stator bölgesine olan bağlantısı çok önem arz etmektedir. Çünkü bu bölgelerde ani basınç kayıpları görülebilmektedir. Buna ek olarak bu bölgelerde akış yönü değişmekte, basınç değişimlerinin de etkisiyle akış türbülanslı bir karakter sergilemektedir. Akış ayrılmaları, girdap oluşumu gibi akış olayları ve hız profilinde ani değişiklikler görülmektedir. Bu yüzden bu geçiş bölgelerinin tasarımı doğru şekilde yapılmalıdır, aksi durumda kavitasyon, gürültü gibi istenmeyen olaylar ortaya çıkacaktır. Sonucunda pompa ömrü azalacak, uzun vadede pompa performansının düşmesine sebep olabilecektir.

Elde edilen sonuçlar belirli bir akışkan ele alınarak bulunmuştur. Farklı özellikteki sıvılar veya gazlar için durum değişiklik gösterebilir. Yeni bir çalışma konusu olarak, paletli pompada gaz transferi üzerine bu çalışmanın benzeri yapılabilir.

Bu konuda yapılabilecek başka bir çalışma da, değişken deplasmanlı paletli pompalarda rotor ve stator arasındaki eksen kaçıklığı formüle edilip sabit bir debi değeri belirlenerek pompa devir sayısı ile eksen kaçıklığının ayarlanması simüle edilebilir.

KAYNAKLAR

[1] Li, T., 2008. Design and Control of a Variable Displacement Vane Pump For Valveless Hydraulic Actuation, Doctoral Thesis, Vanderbilt University, Tennessee.

[2] Karmel, A., (1986). "A Study of the Internal Forces in a Variable-Displacement Vane-Pump – Part I: A Theoretical Analysis", Journal of Fluid Engineering, 108:227-232.

[3] Cavallari, M., 2010. A Lumped Parameter Model for the Pressure and Vibration Analysis of Variable Displacement Vane Pumps, Doctoral Thesis, University of Ferrara, Ferrara.

[4] Wang, L., 2011. Adaptive Control of Variable Displacement Pumps, Doctoral Thesis, Georgia Institute of Technology, Atlanta.

[5] Petrovic, R., (2009). "Mathematical Modeling and Experimantal Verification of Operating Parameters of Vane Pump with Double Effect", Journal of Mechanical Engineering, 55(1):26-32.

[6] Giuffrida, A. and Lanzafame, R., (2005). "Cam Shape and Theoretical Flow Rate in Balanced Vane Pumps", Mechanism and Machine Theory, 40:353-369.

[7] Elashmawy, M., (2014). "Theoretical Investigation of Friction Forces Between Vane Tip and Cam-Ring in Oil Vane Pumps", International Journal of Science, Technology and Society, 2(5):121-128.

[8] Inaguma, Y., (2010). "Theoretical Analysis of Mechanical Efficiency in Vane Pump", JTEKT Engineering Journal English, 1007E:28-35.

[9] Pan, Z., and Lei, C., (2012). "Theoretical Analysis of Experiment Study of Balanced Type Variable Vane Pump", 2nd International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), 2012, Paris.

[10] Frendo, F., Novi, N. and Lei, C., (2006). "Numerical and Experimental Analysis of Variable Displacement Vane Pumps", International Conference on Tribology, 20-22 September 2006, Parma.

[11] Öztürk, R., (2007). Hidrolik – Pnömatik, 2. Baskı, Yıldız Teknik Üniversitesi Yayınları, İstanbul.

[12] Ay, İ., Hidrolik – Pnömatik Ders Notları, Balıkesir Üniversitesi, Balıkesir.

[13] Çengel, Y. ve Cimbala, J., Çev. Ed. Engin, T., (2008). Fluid Mechanics Fundamentals and Applications: Akışkanlar Mekaniği Temelleri ve Uygulamaları, 1. Baskı, Güven Bilimsel Yayınevi, İzmir.

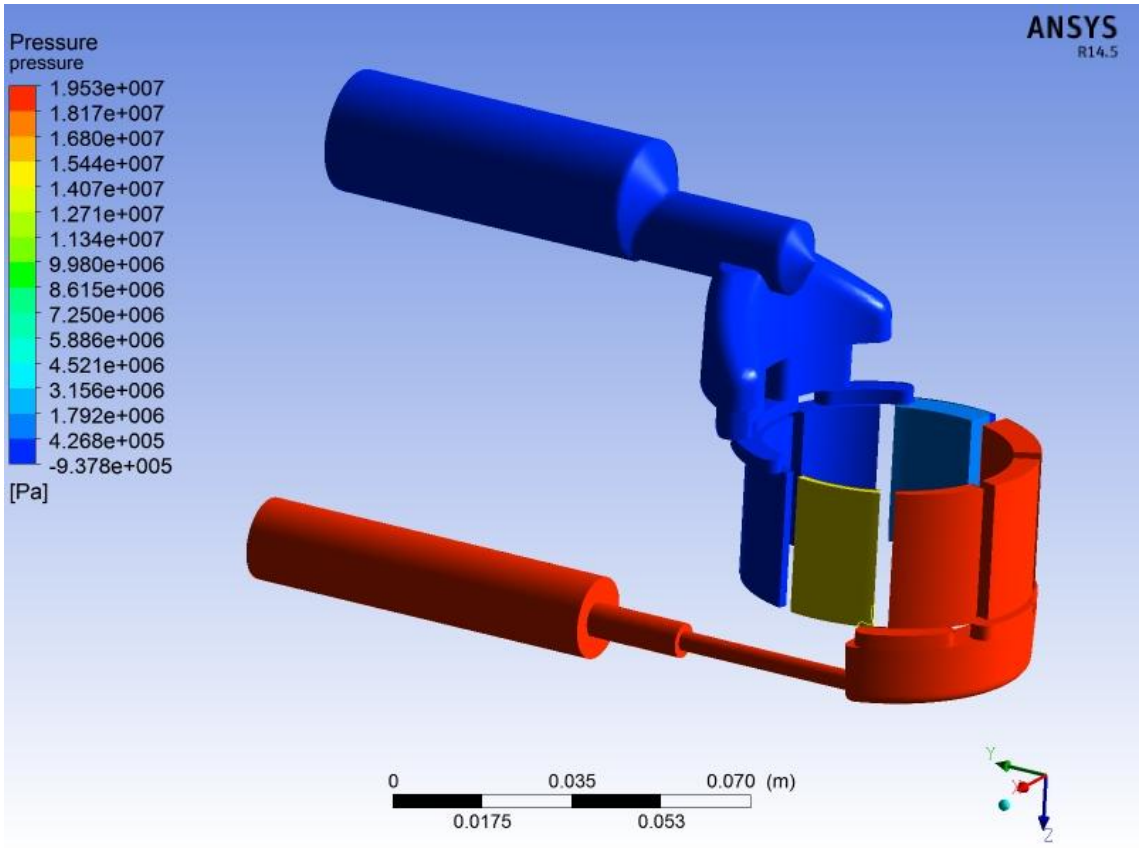
[14] Chung, T., (2002). Computational Fluid Dynamics, 1. Baskı, Cambridge Üniversitesi Yayınları, Cambridge.

[15] Kuzmin , D., Introduction to CFD Lectures, http://www.mathematik.uni-dortmund.de/~kuzmin/cfdintro/cfd.html, 22 Ocak 2015.

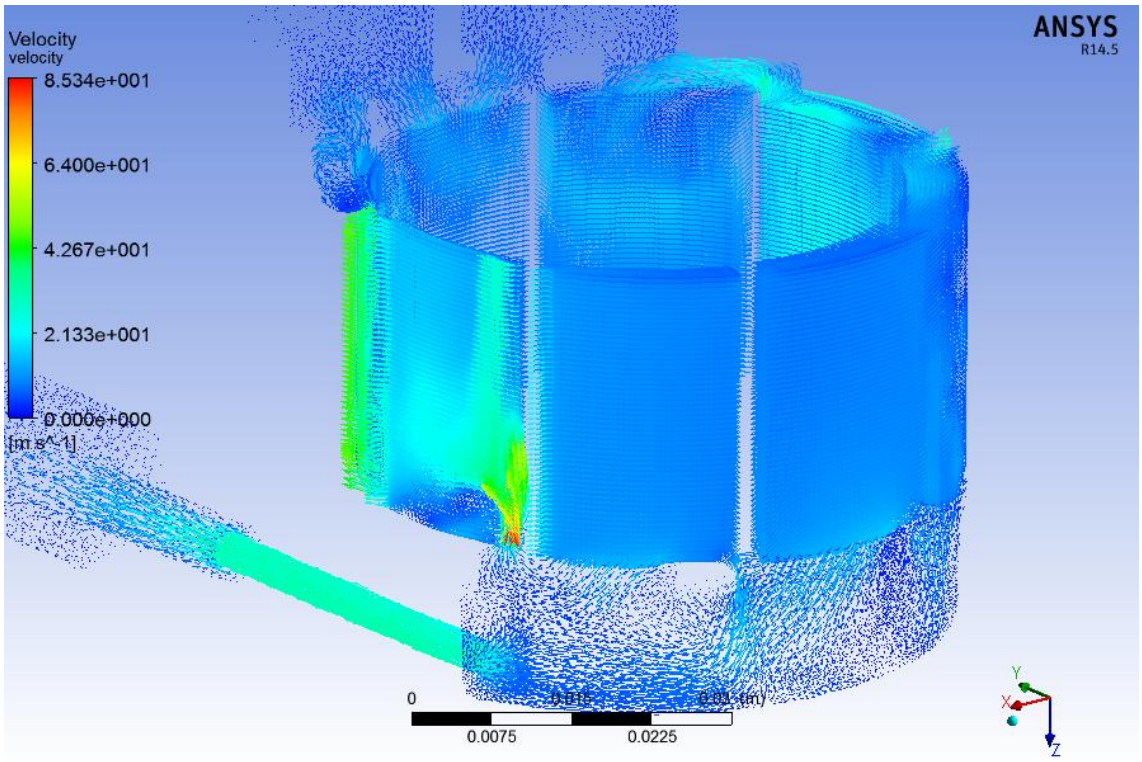
[16] Fluent 6.3 Documentation, http://aerojet.engr.ucdavis.edu/fluenthelp/, 12 Şubat 2015.

8 Kanatlı Pompa Sonuçları

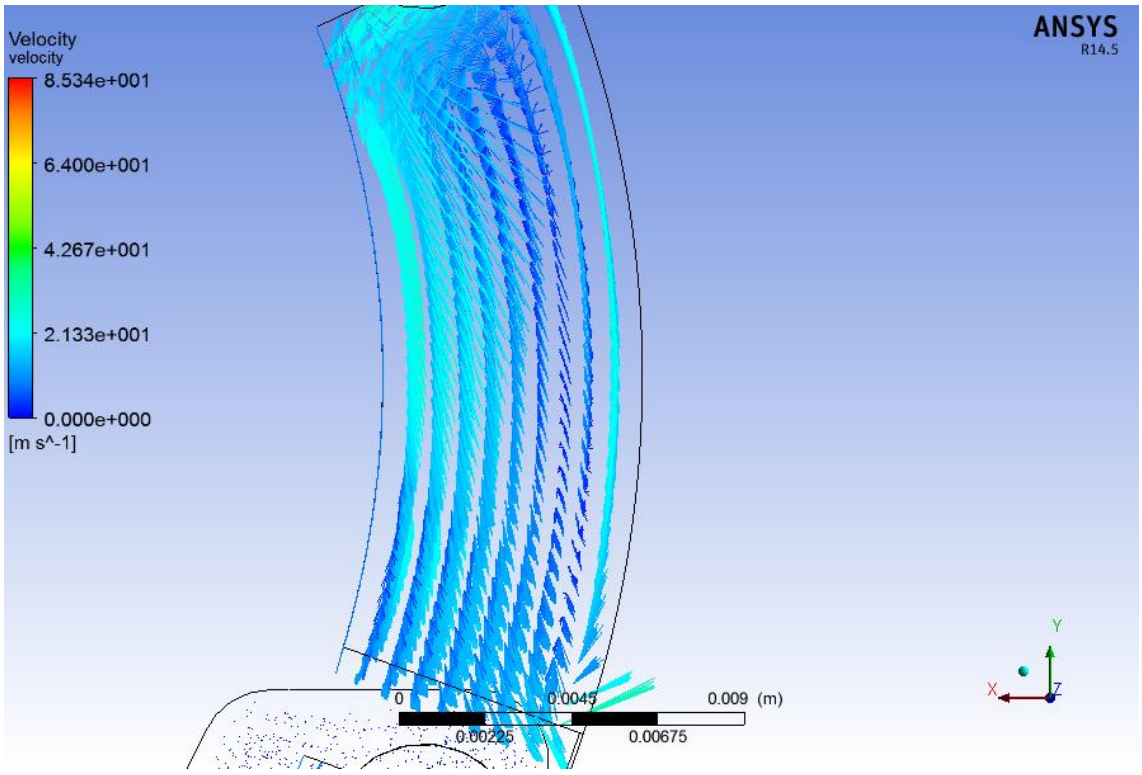
8 kanatlı pompaya ait basınç dağılımları aşağıda gösterilmiştir.



8 kanatlı pompaya ait hız vektörleri aşağıda gösterilmiştir.



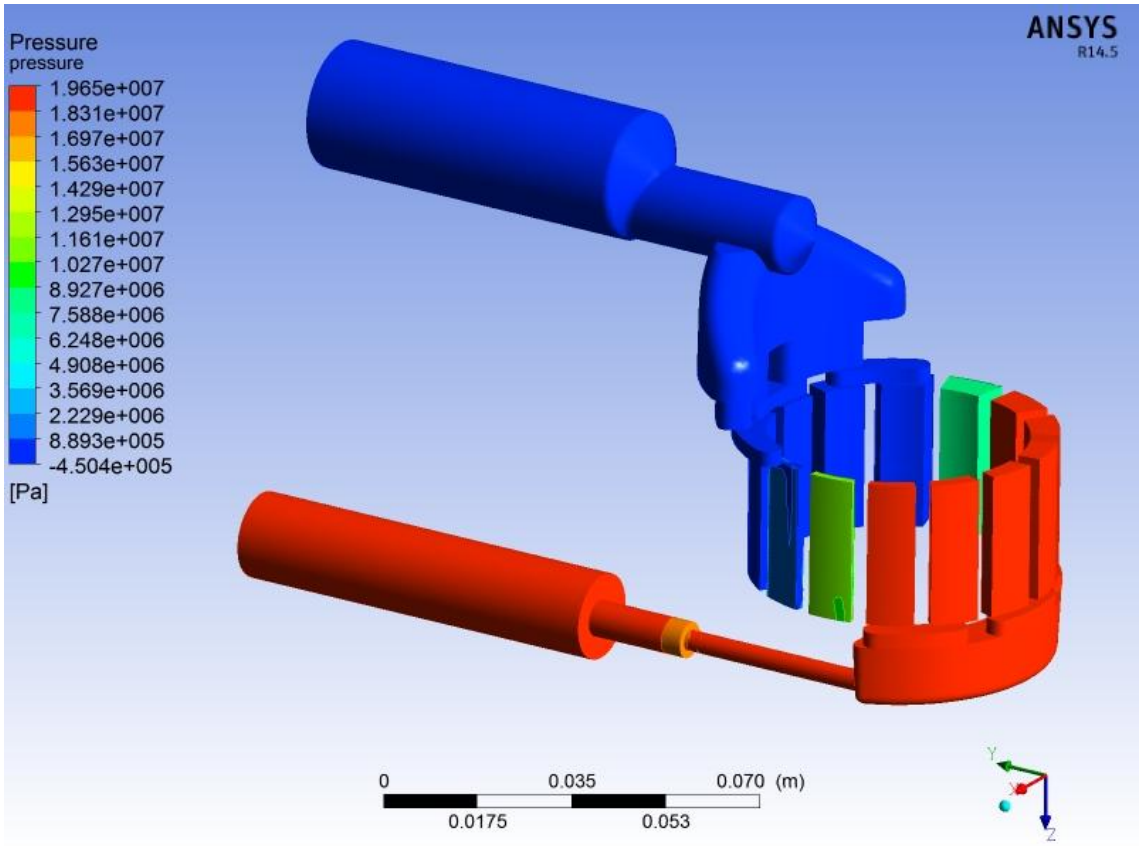
8 kanatlı pompaya ait pompa havuzu hız vektörleri aşağıda gösterilmiştir.



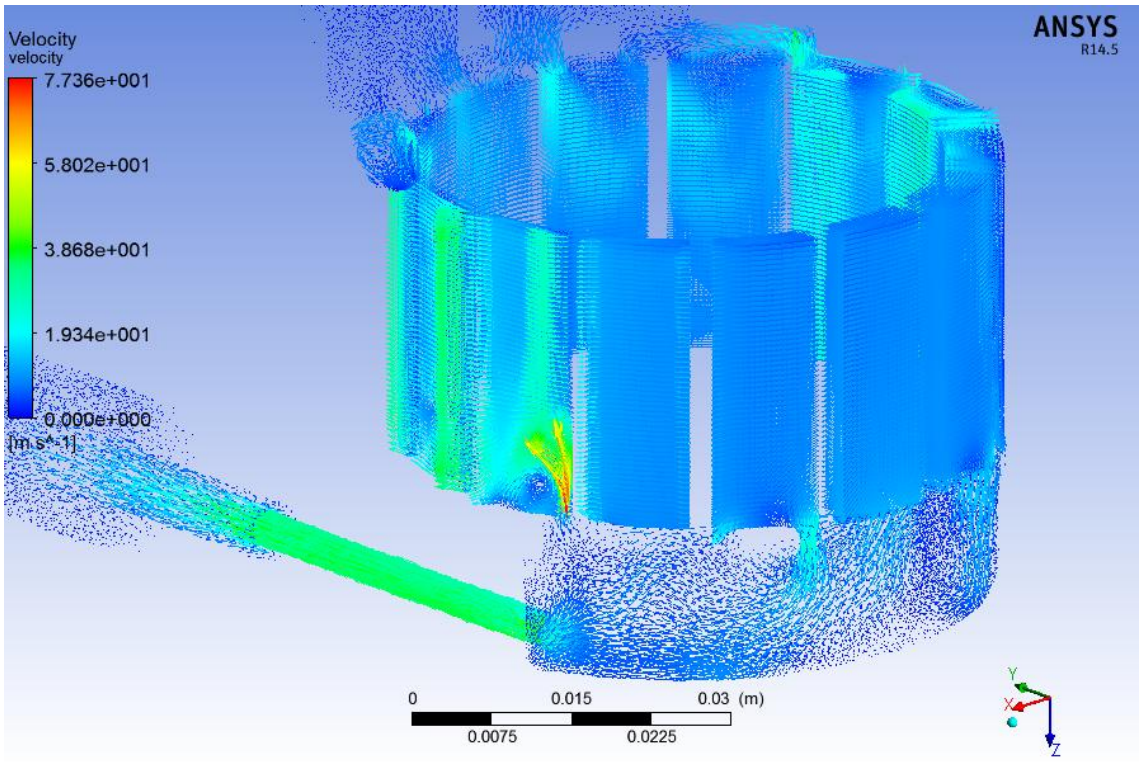
EK-B

14 Kanatlı Pompa Sonuçları

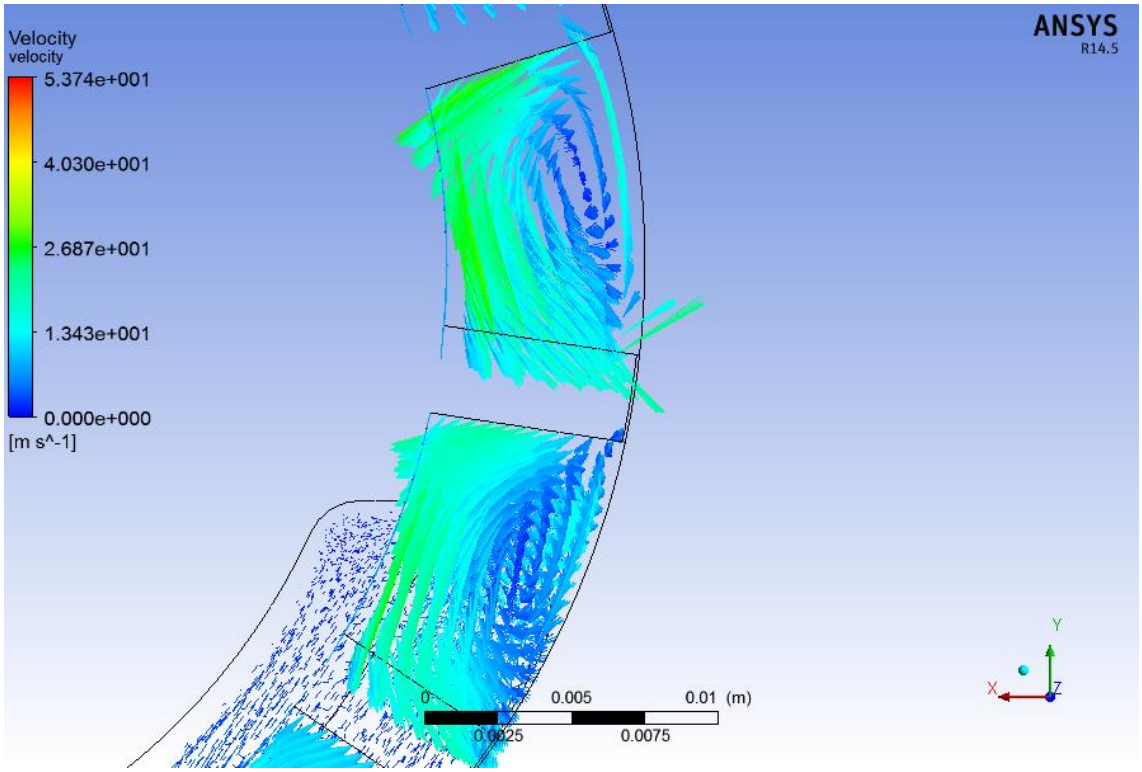
14 kanatlı pompaya ait basınç dağılımları aşağıda gösterilmiştir.



14 kanatlı pompaya ait hız vektörleri aşağıda gösterilmiştir.



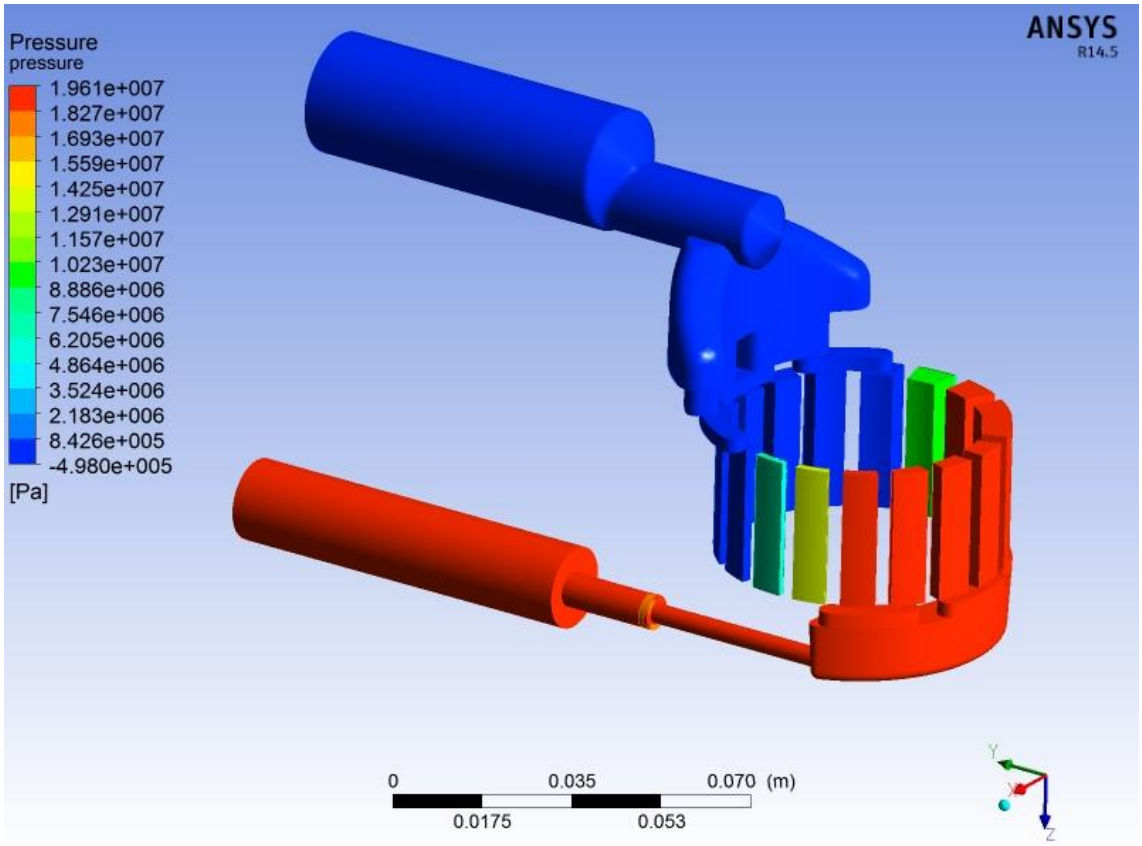
14 kanatlı pompaya ait pompa havuzu hız vektörleri aşağıda gösterilmiştir.



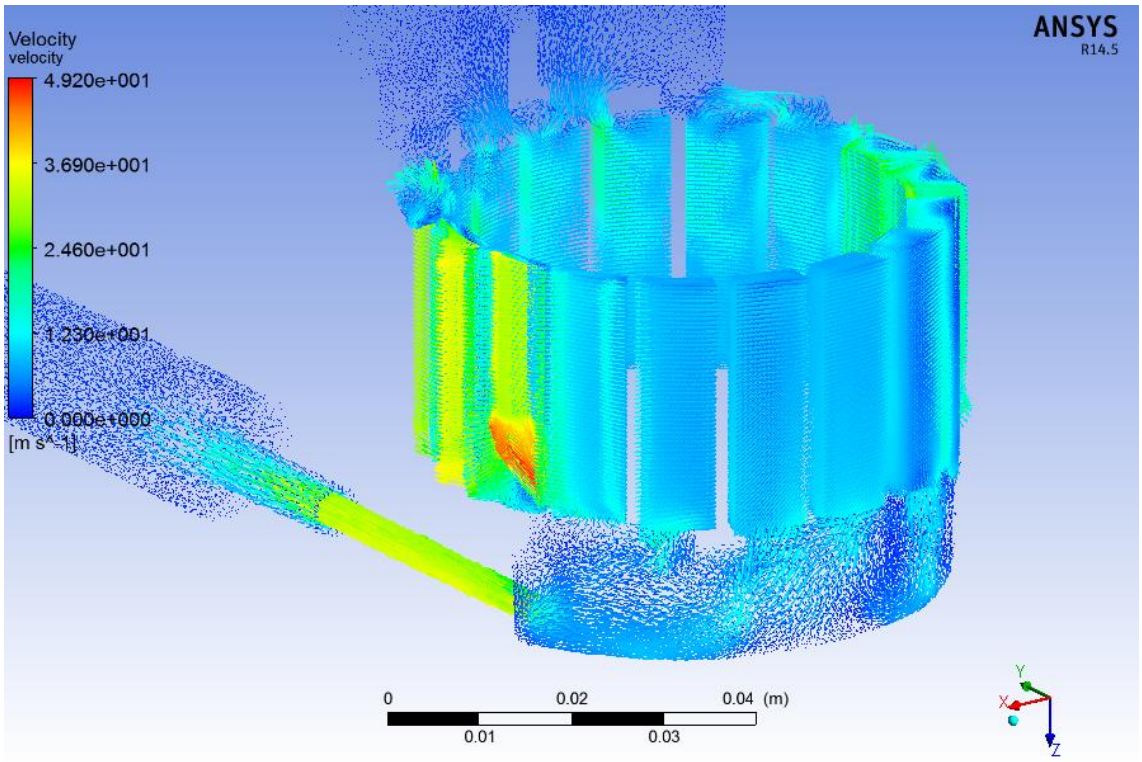
EK-C

17 Kanatlı Pompa Sonuçları

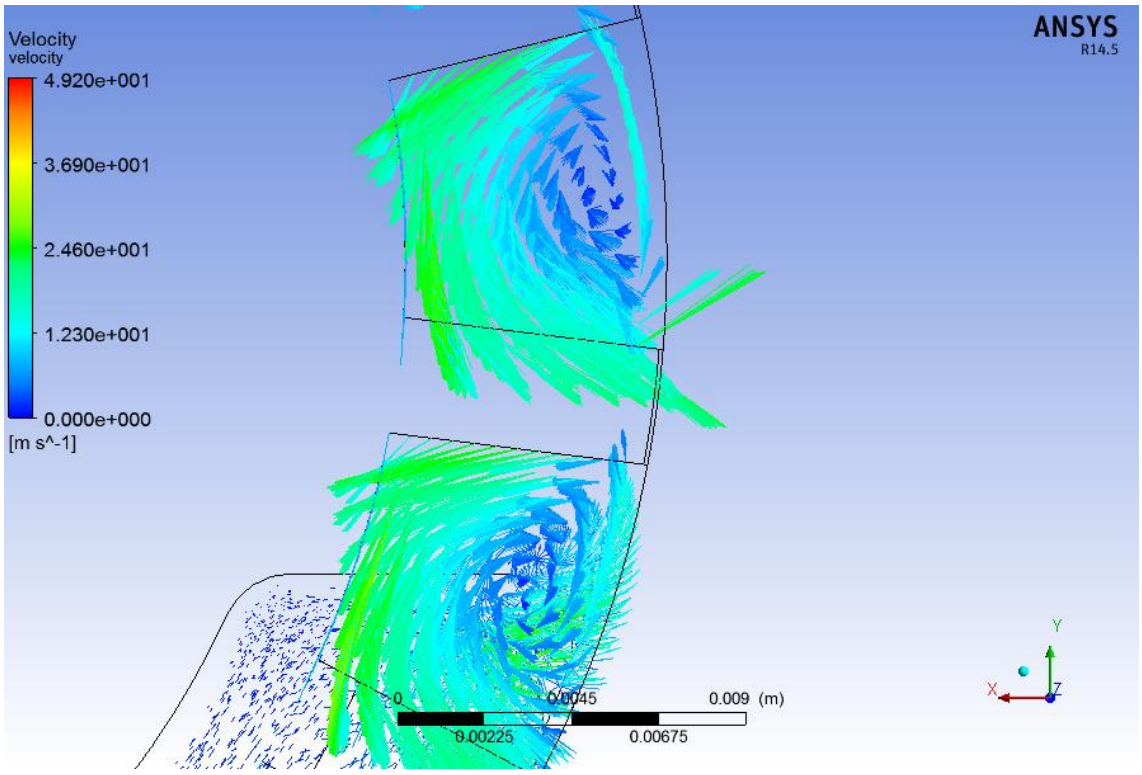
17 kanatlı pompaya ait basınç dağılımları aşağıda gösterilmiştir.



17 kanatlı pompaya ait hız vektörleri aşağıda gösterilmiştir.

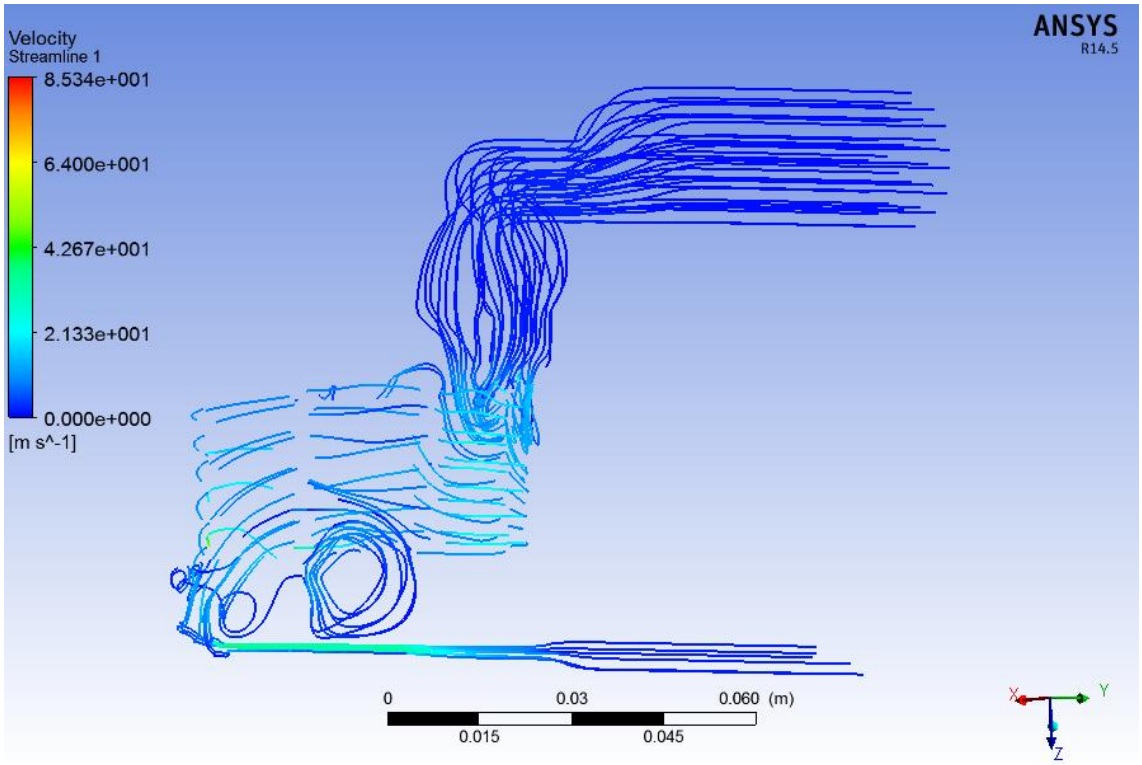


17 kanatlı pompaya ait pompa havuzu hız vektörleri aşağıda gösterilmiştir.

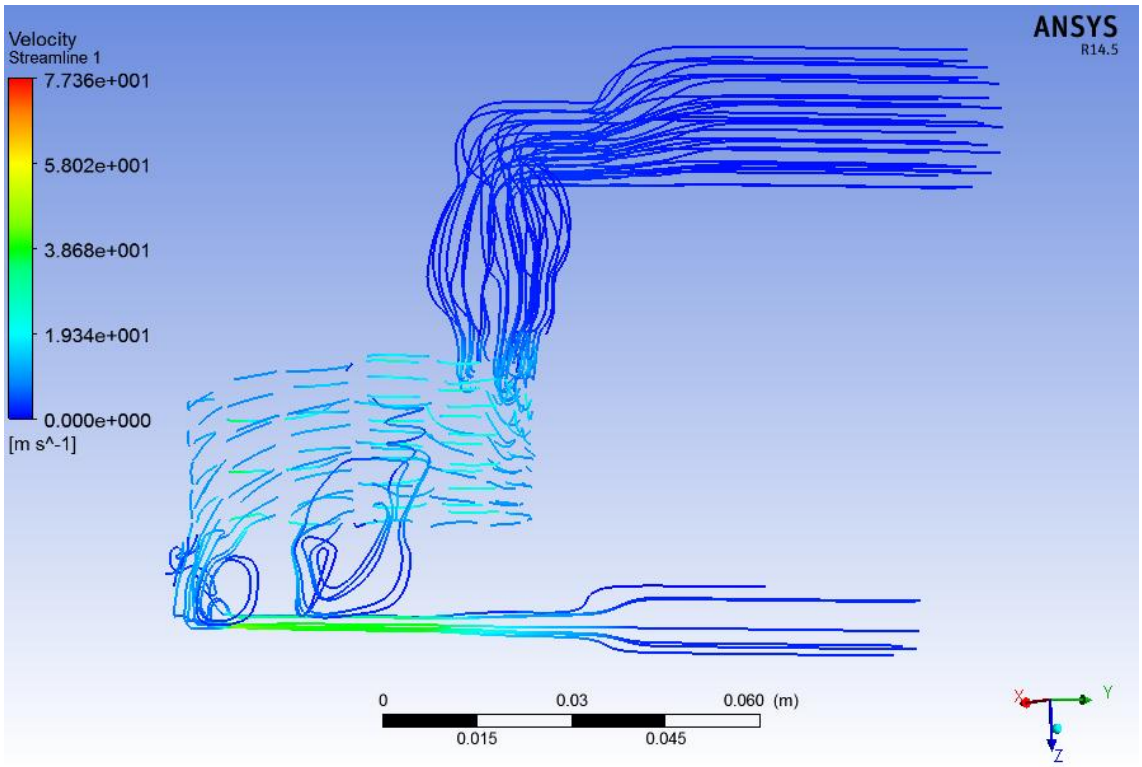


Çeşitli Kanat Sayılarına Göre Akım Çizgileri

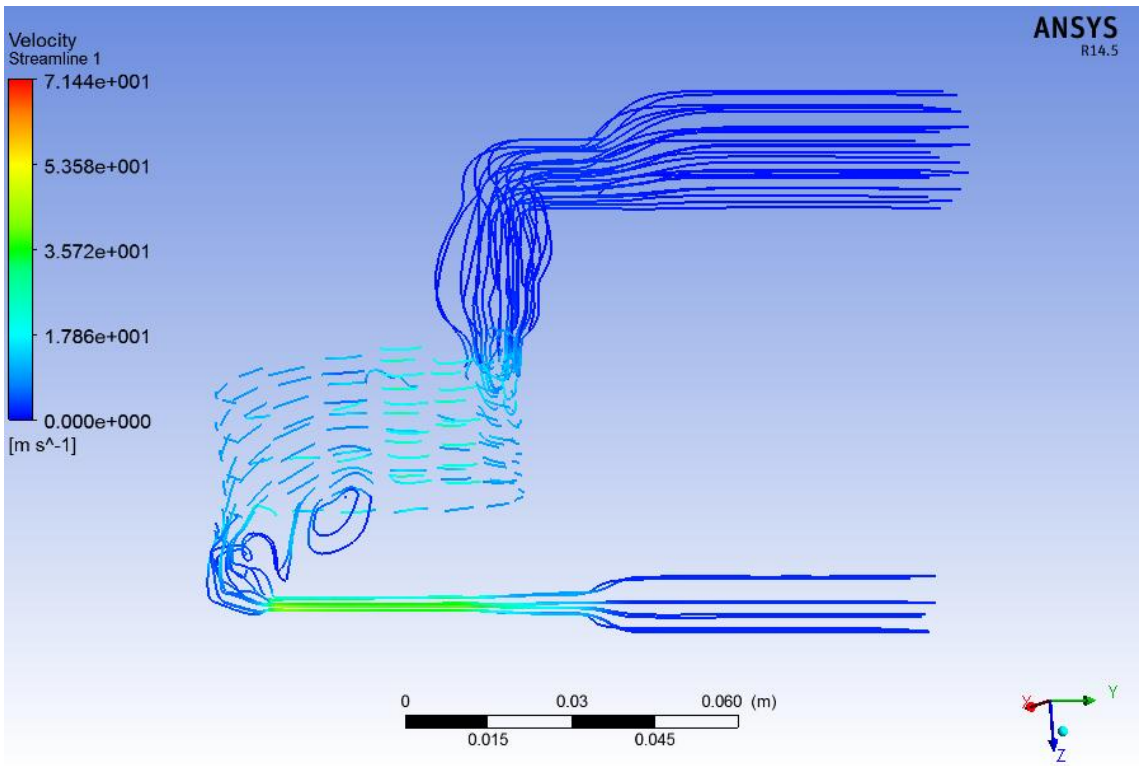
8 kanatlı pompaya ait akım çizgileri aşağıda gösterilmiştir.



14 kanatlı pompaya ait akım çizgileri aşağıda gösterilmiştir.



17 kanatlı pompaya ait akım çizgileri aşağıda gösterilmiştir.



Kullanıcı Tanımlı Fonsiyon (UDF) Kodları

```
# include "udf.h"
# include "dynamesh_tools.h"
# define DEBUG 0
#if RP_DOUBLE
#  define REAL_FMT "%le"
#else
#  define REAL_FMT "%e"
#endif


extern void find_intersection(real (*)[2], int , real, real, real, real, real *, real *);
extern void find_appp(real *, real *, real *, real, real, real, real (*)(real *, real *, real), real (*)(real *, real *, real));
extern void find_app_rigid(real *, real *, real *, real);

/* extern real special_inner_radius(real z); */

/* Outer contour shape */
enum define_shape
{
  circle, user_defined, special_circle, special_user_defined
}OUTER_SHAPE, INNER_SHAPE;

/* Number of vanes, core ID and gap ID */
static int N_VANE;

/* single core and single gap flag*/
static int MULTIPLE_CORE, MULTIPLE_GAP;

/* Core ID and gap ID */
static int * CORE_ID, * GAP_ID;

/* Rotating inner profile flag */
static int INNER_TIME_DEP_PROFILE;

/* rotor rpm */
static real RPM;

/* Outer circle radius, inner circle radius, half vane width, gap size and offset */
static real R_OUTER, R_INNER, VANE_WIDTH, GAP_SIZE, DELTA;

/* initial vane angle */
static real INIT_VANE_ANGLE;

/* Flag for vane spacing */
static int NON_EQUALLY_SPACED_VANE;

/* angle for sectors */
```

```
static real * ANGLE;

/* Number of layers */
int Num_Layers=20;     /* hard coded number for vane */

enum shrink
{
  w_shrink, wo_shrink
};

struct shape
{
 enum define_shape shape_type;

 enum shrink shrink_y_or_n;

 real center[2];
 real radius;

 real (*data)[2];
 int number_of_data;

 int flag_time_dep_profile;

}pump_core_outer_shape, pump_core_inner_shape, pump_gap_outer_shape, pump_gap_inner_shape;

enum UDM
{
  UDM_sector
};

/* Allocate memory for 1d int array with dimention of n_x */
int * alloc_1d_int_array(int n_x)
{
   int *x;

   if(!(x = (int *) calloc(n_x, sizeof(int))))
     {
   Message0("Memory y allocation failure-aboring\n\n");
   exit(0);
  }

return x;
}

/* Allocate memory for 1d real array with dimention of n_x */
real * alloc_1d_real_array(int n_x)
{
   real *x;

   if(!(x = (real *) calloc(n_x, sizeof(real))))
     {
   Message0("Memory y allocation failure-aboring\n\n");
   exit(0);
  }

return x;
}


static real my_atan2(real y, real x)
{
 real result;

 result=atan2(y,x);
 if(atan2(y,x)<0)
```

```
 result = atan2(y,x) + 2*M_PI;

 return result;
}


/* Function used to calcuate op for the gap.  Please refer to note page 4 for the definition. */

static void f_Op_gap(real * xop, real * yop, real xa, real ya, real time, real sector)
{
 real alfa, h;

 alfa = time*RPM*M_PI/30+sector*2*M_PI/N_VANE;

 h=cos(alfa)*ya-sin(alfa)*xa;

 if(fabs(cos(alfa))>0.1)
  {
  *xop = 0.5*xa;
  *yop = (sin(alfa)*0.5*xa+h)/cos(alfa);
  }
 else
  {
     *xop = (cos(alfa)*0.5*ya-h)/sin(alfa);
  *yop = 0.5*ya;
  }

 return;
}

/* Function used to calcuate op.  Please refer to note page 3 for the definition. */

static void f_Op_core(real * x, real * y, real time, real sector)
{
 real alfa;
 int i;
 real angle;

 angle = 0;
 if(NON_EQUALLY_SPACED_VANE)
  {
   for(i=0; i<=sector-1; i++)
     {
  angle +=ANGLE[i-1];
  }

    alfa = time*RPM*M_PI/30+angle+INIT_VANE_ANGLE*M_PI/180;

    *x = 0.5*VANE_WIDTH*(cos(alfa)+cos(alfa+ANGLE[(int) (sector+0.1)]))/sin(ANGLE[(int) (sector+0.1)]);
    *y = 0.5*VANE_WIDTH*(sin(alfa)+sin(alfa+ANGLE[(int) (sector+0.1)]))/sin(ANGLE[(int) (sector+0.1)]);
  }
 else
  {
   alfa = time*RPM*M_PI/30+sector*2*M_PI/N_VANE+INIT_VANE_ANGLE*M_PI/180;

    *x = 0.5*VANE_WIDTH*(cos(alfa)+cos(alfa+2*M_PI/N_VANE))/sin(2*M_PI/N_VANE);
    *y = 0.5*VANE_WIDTH*(sin(alfa)+sin(alfa+2*M_PI/N_VANE))/sin(2*M_PI/N_VANE);
  }


 return;
}

/* Rotate a profile around the origin */
static void rotate_profile(real (* orig_profile)[2], real (* final_profile)[2],int no_data, real theta)
{
```

```
 int i;
 real origin[2];

 origin[0]=origin[1]=0;
 for(i=0; i<=no_data-1; i++)
   {
    find_app_rigid(orig_profile[i], final_profile[i], origin, theta);
   }
}

/* Generic function to calcualte the distance between point a and c.  Please refer to note page 2.
  The inputs are points a, b, center of the circle (x0, y0), and the circle radius.  The h and H
  for this application are all from this function.
*/
static real f_h(real xa, real ya, real xb, real yb, real z_coord, struct shape * contour_shape, real time)
{
 if (contour_shape->shape_type==circle)
   {
    real t1, t2, a, b, c, t, x0, y0, radius;

    x0 = contour_shape->center[0];
    y0 = contour_shape->center[1];
    radius = contour_shape->radius;

    a=pow(xb-xa,2)+pow(yb-ya,2);
    b=2*(xa-x0)*(xb-xa)+2*(ya-y0)*(yb-ya);
    c=pow(xa-x0,2)+pow(ya-y0,2)-radius*radius;

    t1 = (-b+sqrt(b*b-4*a*c))/(2*a);
    t2 = (-b-sqrt(b*b-4*a*c))/(2*a);

    if (t1>0&&t2<0)
    t = t1;
    else if (t2>0&&t1<0)
    t = t2;
    else
    {
     Message("\nSomething wrong with f_h. t1=%7.2f t2=%7.2f", t1, t2);
     Message("\nxa=%11.3e ya=%11.3e xb=%11.3e yb=%11.3e x0=%10.3e y0=%10.3e radius=%10.3e", xa, ya, xb, yb, x0, y0,
radius);

     t = 1;
    }

    return t*sqrt((pow(yb-ya,2)+pow(xb-xa,2)));
   }
  else if (contour_shape->shape_type==user_defined)
   {
 real x, y;

     if(contour_shape->flag_time_dep_profile)
      {
       real (*final_profile)[2];

       if(!(final_profile=(real (*)[2]) calloc(contour_shape->number_of_data, 2*sizeof(real))))
        {
  Message0("\nMemory allocatoin failure-aborting!!\n");
  exit(0);
     }
       rotate_profile(contour_shape->data, final_profile, contour_shape->number_of_data, time*RPM*M_PI/30);

       find_intersection(final_profile, contour_shape->number_of_data, xa, ya, xb, yb, &x, &y);

       free(final_profile);
  }
    else
```

```
    {
  find_intersection(contour_shape->data, contour_shape->number_of_data, xa, ya, xb, yb, &x, &y);
    }

   if(contour_shape->shrink_y_or_n==w_shrink)
    {
      return sqrt(pow(y-ya,2)+pow(x-xa,2))-GAP_SIZE;
  }
   else
    {
      return sqrt(pow(y-ya,2)+pow(x-xa,2));
  }
}
 else if (contour_shape->shape_type==special_circle)
  {
   real t1, t2, a, b, c, t, x0, y0, radius;

   x0 = 0;
   y0 = 0;

   /* radius = special_inner_radius(z_coord); */
   radius = 1;  /* Hardwire option to get rid of the above function */

   a=pow(xb-xa,2)+pow(yb-ya,2);
   b=2*(xa-x0)*(xb-xa)+2*(ya-y0)*(yb-ya);
   c=pow(xa-x0,2)+pow(ya-y0,2)-radius*radius;

   t1 = (-b+sqrt(b*b-4*a*c))/(2*a);
   t2 = (-b-sqrt(b*b-4*a*c))/(2*a);

   if (t1>0&&t2<0)
   t = t1;
   else if (t2>0&&t1<0)
   t = t2;
   else
   {
    Message("\nSomething wrong with f_h. t1=%7.2f t2=%7.2f", t1, t2);
    Message("\nxa=%11.3e ya=%11.3e xb=%11.3e yb=%11.3e x0=%10.3e y0=%10.3e radius=%10.3e", xa, ya, xb, yb, x0, y0,
radius);

    t = 1;
   }

   return t*sqrt((pow(yb-ya,2)+pow(xb-xa,2)));
}
 else if (contour_shape->shape_type==special_user_defined)
  {
real x, y, radius;

   if(contour_shape->flag_time_dep_profile)
    {
      real (*final_profile)[2];

      if(!(final_profile=(real (*)[2]) calloc(contour_shape->number_of_data, 2*sizeof(real))))
       {
 Message0("\nMemory allocatoin failure-aborting!!\n");
 exit(0);
    }
      rotate_profile(contour_shape->data, final_profile, contour_shape->number_of_data, time*RPM*M_PI/30);

      find_intersection(final_profile, contour_shape->number_of_data, xa, ya, xb, yb, &x, &y);

      free(final_profile);
  }
   else
    {
```

```
  find_intersection(contour_shape->data, contour_shape->number_of_data, xa, ya, xb, yb, &x, &y);
     }

    if(contour_shape->shrink_y_or_n==w_shrink)
     {
        /* radius = special_inner_radius(z_coord); */
        radius = 1;  /* Hardwire option to get rid of the above function */
        return sqrt(pow(y-ya,2)+pow(x-xa,2))-GAP_SIZE+radius;
  }
    else
     {
        /* radius = special_inner_radius(z_coord); */
        radius = 1;  /* Hardwire option to get rid of the above function */
        return sqrt(pow(y-ya,2)+pow(x-xa,2))+radius;
  }
}
  else
   {
Message0("\nwrong type-aborting!!!\n");
exit(0);
}
}

static real f_h_core(real * op, real *ap, real time)
{
 return f_h(op[0], op[1], ap[0], ap[1], ap[2], &pump_core_inner_shape, time);
}

static real f_H_core(real * op, real *ap, real time)
{
 real not_used_var=0;
 return f_h(op[0], op[1], ap[0], ap[1], not_used_var, &pump_core_outer_shape, time);
}

static real f_h_gap(real * op, real *ap, real time)
{
 real not_used_var=0;
 return f_h(op[0], op[1], ap[0], ap[1], not_used_var, &pump_gap_inner_shape, time);
}

static real f_H_gap(real * op, real *ap, real time)
{
 real not_used_var=0;
 return f_h(op[0], op[1], ap[0], ap[1], not_used_var, &pump_gap_outer_shape, time);
}

/* Functions used to calcuate a new position for vane pump core */

DEFINE_GRID_MOTION(vane_pump_core, domain, dt, time, dtime)
{
  cell_t c;
  Thread *tc = DT_THREAD ((Dynamic_Thread *)dt);
  int n;
  Node *v;
  real ap[3], op[2], appp[2];

  /* set deforming flags */
  SET_DEFORMING_THREAD_FLAG (tc);

  begin_c_loop(c, tc)
   {
     c_node_loop(c, tc, n)
      {
        v = C_NODE(c, tc, n);
        if (NODE_POS_NEED_UPDATE(v))
         {
```

```
          NODE_POS_UPDATED(v);

        ap[0]=NODE_X(v);
        ap[1]=NODE_Y(v);
        ap[2]=NODE_Z(v);

          f_Op_core(op, op+1, time-dtime, C_UDMI(c, tc, UDM_sector));
          find_appp(appp, ap, op, time-dtime, dtime, RPM*M_PI/30*dtime, f_h_core, f_H_core);

        NODE_X(v)=appp[0];
        NODE_Y(v)=appp[1];
         }
       }
     Update_Cell_Metrics (c, tc);
   }
  end_c_loop (c, tc);

}

/* Functions used to calcuate a new position for vane pump gap */

DEFINE_GRID_MOTION(vane_pump_gap, domain, dt, time, dtime)
{
  cell_t c;
  Thread *tc = DT_THREAD ((Dynamic_Thread *)dt);
  int n;
  Node *v;
  real ap[3], op[2], appp[2];

  /* set deforming flags */
  SET_DEFORMING_THREAD_FLAG (tc);

  begin_c_loop(c, tc)
   {
     c_node_loop(c, tc, n)
      {
        v = C_NODE(c, tc, n);
        if (NODE_POS_NEED_UPDATE(v))
         {
           NODE_POS_UPDATED(v);

        ap[0]=NODE_X(v);
        ap[1]=NODE_Y(v);
        ap[2]=NODE_Z(v);

          f_Op_gap(op, op+1, ap[0], ap[1], time-dtime, C_UDMI(c, tc, UDM_sector));
          find_appp(appp, ap, op, time-dtime, dtime, RPM*M_PI/30*dtime, f_h_gap, f_H_gap);

        NODE_X(v)=appp[0];
        NODE_Y(v)=appp[1];
         }
       }
     Update_Cell_Metrics (c, tc);
   }
  end_c_loop (c, tc);

}

/* Functions used to calcuate a new position for vane pump core top face (for F6.3 only) */

/* Functions used to calcuate a new position for vane pump gap top face (for F6.3 only) */

DEFINE_GRID_MOTION(walls, domain, dt, time, dtime)
{

}
```

```
static void intialize_one_shape(struct shape * one_shape,  enum define_shape shape_type, enum shrink shrink_y_or_n,
real * center, real radius,
                        char * file_name, int flag_time_dep_profile)
{
 one_shape->shape_type=shape_type;
 one_shape->flag_time_dep_profile=flag_time_dep_profile;
 if (one_shape->shape_type==circle)
   {
one_shape->center[0]=center[0];
one_shape->center[1]=center[1];
one_shape->radius=radius;
}
 else if (one_shape->shape_type==user_defined)
   {
    FILE *fp_data;
    int i, j;

    one_shape->shrink_y_or_n=shrink_y_or_n;

    #if PARALLEL
     if(I_AM_NODE_HOST_P)
    #endif
      {
       if(!(fp_data=fopen(file_name,"r")))
        {
          Message0("\nCan not open file %s -aborting!!", file_name);
          exit(0);
        }

       fscanf(fp_data, "%d", &(one_shape->number_of_data));
      }

    host_to_node_int_1(one_shape->number_of_data);

    if(!(one_shape->data=(real (*)[2]) calloc(one_shape->number_of_data, 2*sizeof(real))))
     {
  Message0("\nMemory allocatoin failure-aborting!!\n");
  exit(0);
  }

    for(i=0; i<one_shape->number_of_data; i++)
     for(j=0; j<2; j++)
     {
        #if PARALLEL
         if(I_AM_NODE_HOST_P)
        #endif
          {
             #if RP_DOUBLE
       fscanf(fp_data, "%le", &(one_shape->data[i][j]));
             #else
       fscanf(fp_data, "%e", &(one_shape->data[i][j]));
             #endif
          }
  }

    host_to_node_real(&(one_shape->data[0][0]), 2*one_shape->number_of_data);

    #if PARALLEL
     if(I_AM_NODE_HOST_P)
    #endif
      {
       fclose(fp_data);
      }
}
 else if (one_shape->shape_type==special_circle)
```

```
  {

  }
 else if (one_shape->shape_type==special_user_defined)
  {
   FILE *fp_data;
   int i, j;

   one_shape->shrink_y_or_n=shrink_y_or_n;

   #if PARALLEL
    if(I_AM_NODE_HOST_P)
   #endif
     {
      if(!(fp_data=fopen(file_name,"r")))
       {
         Message0("\nCan not open file %s -aborting!!", file_name);
         exit(0);
       }

      fscanf(fp_data, "%d", &(one_shape->number_of_data));
     }

   host_to_node_int_1(one_shape->number_of_data);

   if(!(one_shape->data=(real (*)[2]) calloc(one_shape->number_of_data, 2*sizeof(real))))
    {
  Message0("\nMemory allocatoin failure-aborting!!\n");
  exit(0);
  }

   for(i=0; i<one_shape->number_of_data; i++)
    for(j=0; j<2; j++)
    {
       #if PARALLEL
        if(I_AM_NODE_HOST_P)
       #endif
         {
           #if RP_DOUBLE
      fscanf(fp_data, "%le", &(one_shape->data[i][j]));
           #else
      fscanf(fp_data, "%e", &(one_shape->data[i][j]));
           #endif
         }
  }

   host_to_node_real(&(one_shape->data[0][0]), 2*one_shape->number_of_data);

   #if PARALLEL
    if(I_AM_NODE_HOST_P)
   #endif
     {
      fclose(fp_data);
     }
  }
 else
  {
Message0("\nWrong shape-aborting!!!\n");
exit(0);
}

}

static void initialize_shape(void)
{
 real center[2], radius;
```

```
 center[0]=-DELTA;
 center[1]=0;
 radius=R_OUTER;
 intialize_one_shape(&pump_core_outer_shape, OUTER_SHAPE, wo_shrink, center, radius, "data_outer.txt", 0); /*hard
coded no time dep prof*/

 center[0]=0;
 center[1]=0;
 radius=R_INNER;
 intialize_one_shape(&pump_core_inner_shape, INNER_SHAPE, wo_shrink, center, radius, "data_inner.txt",
INNER_TIME_DEP_PROFILE);

 center[0]=-DELTA;
 center[1]=0;
 radius=R_OUTER;
 intialize_one_shape(&pump_gap_outer_shape, OUTER_SHAPE, wo_shrink, center, radius, "data_outer.txt", 0);/*hard
coded no time dep prof*/

 center[0]=-DELTA;
 center[1]=0;
 radius=R_OUTER-GAP_SIZE;
 intialize_one_shape(&pump_gap_inner_shape, OUTER_SHAPE, w_shrink, center, radius, "data_outer.txt", 0);/*hard
coded no time dep prof*/

 return;
}

static int sector_number(real angle)
{
 int i;
 real total_angle;

 i = 0;
 total_angle = ANGLE[0];

 while(angle>total_angle)
  {
  i++;
  total_angle += ANGLE[i];
  }

 return i;
}

static void init_one_core_cell_zone(Thread * tc)
{
 cell_t c;
 Node *v;
 int j;
 real angle, init_vane_angle_rad;

 begin_c_loop_int(c, tc)
  {
    v = C_NODE(c, tc, 0);

    angle = my_atan2(NODE_Y(v), NODE_X(v));
    init_vane_angle_rad = INIT_VANE_ANGLE*M_PI/180;

    angle -= init_vane_angle_rad;

    if (angle <0)
      angle += 2*M_PI;

    if(NON_EQUALLY_SPACED_VANE)
     {
```

```
      C_UDMI(c, tc, UDM_sector) = sector_number(angle);
  }
   else
    {
   j = angle/(2*M_PI/N_VANE);

      C_UDMI(c, tc, UDM_sector) = j;
  }
  }
 end_c_loop_int(c, tc)
}

static void init_one_gap_cell_zone(Thread * tc)
{
 cell_t c;
 Node *v;
 int j;
 real angle, init_vane_angle_rad;

 begin_c_loop_int(c, tc)
  {
v = C_NODE(c, tc, 0);

   angle = my_atan2(NODE_Y(v),NODE_X(v));
   init_vane_angle_rad = INIT_VANE_ANGLE*M_PI/180;

   angle -= init_vane_angle_rad;

   if (angle <0)
     angle += 2*M_PI;

if (fabs(angle-2*M_PI)<2*M_PI/N_VANE*0.3)
angle = 0;

j = (angle+2*M_PI/N_VANE*0.5)/(2*M_PI/N_VANE);

C_UDMI(c, tc, UDM_sector) = j;
  }
 end_c_loop_int(c, tc)
}

/* Initialization function is used to calculate sector number */

DEFINE_INIT(init_sector, domain)
{
 Thread *tc;
 int i;

 /* Initialization can only be done at the beginning */
 Message0("\n\n*********************************************************************************");
 Message0("\n\n Warning: you are initializing the flow.  For the vane pump application,");
 Message0("\n you can only do this when the mesh is at the initial position.  Otherwise,");
 Message0("\n initialization may cause mesh motion failure!!!\n");
 Message0("\n After initialization, please display the UDM-0 using cell value to verify");
 Message0("\n that each chamber has distinct value (DIFFERENT SINGLE COLOR FOR EACH CHAMBER!\n");
 Message0("\n*********************************************************************************\n");

 /* Initialize the Core sector number*/

 if(MULTIPLE_CORE)
  {
   for(i=0; i<N_VANE; i++)
    {
     tc=Lookup_Thread(domain, CORE_ID[i]);
     init_one_core_cell_zone(tc);
    }
```

```
  }
 else
  {
    tc=Lookup_Thread(domain, CORE_ID[0]);
    init_one_core_cell_zone(tc);
  }

 /* Initialize the Gap sector number*/

 if (GAP_ID[0] > 0)
  {
   if(MULTIPLE_GAP)
    {
     for(i=0; i<N_VANE; i++)
      {
     tc=Lookup_Thread(domain, GAP_ID[i]);
        init_one_gap_cell_zone(tc);
      }
    }
   else
    {
      tc=Lookup_Thread(domain, GAP_ID[0]);
      init_one_gap_cell_zone(tc);
    }
  }
}

/* print the outer contour for user_defined shape_type*/

static void print_one_shape(struct shape * one_shape)
{
 Message0("shape type: %d\n", one_shape->shape_type);

 if((one_shape->shape_type))
   Message0("shrink: %d\n", one_shape->shrink_y_or_n);

 if (one_shape->shape_type==circle)
  {
    Message0("center:%10.2e %10.2e radius:%10.2e\n", one_shape->center[0], one_shape->center[1], one_shape-
>radius);
  }
 else if(one_shape->shape_type==user_defined || one_shape->shape_type==special_user_defined)
  {

   Message0("number of data: %d\n", one_shape->number_of_data);

#if DEBUG

   int i, j;

   for(i=0; i<one_shape->number_of_data; i++)
    {
     for(j=0; j<2; j++)
       {
    Message0("%12.4e", one_shape->data[i][j]);
    }
     Message0("\n");
    }
#endif
  }
 else if(one_shape->shape_type==special_circle)
  {
  Message0("\nSpecial Inner Circle Shape\n");
  }
 else if(one_shape->shape_type==special_user_defined)
  {
```

```
 Message0("\nSpecial Inner User-Defined Shape\n");
  }
 else
  {
 Message0("\nWrong type-aborting!!\n"); exit(0);
  }
}

/* Only node 0 will execute ON_DEMAND udf */

DEFINE_ON_DEMAND(print_info)
{
  int i;
  real total_angle;

  Message0("\n************************** General info **************************\n");

  Message0("\nOUTER_SHAPE: %d\n", OUTER_SHAPE);
  Message0("INNER_SHAPE: %d\n", INNER_SHAPE);
  Message0("N_VANE: %d\n", N_VANE);

  Message0("MULTIPLE_CORE: %d\n", MULTIPLE_CORE);
  if(MULTIPLE_CORE)
 {
 Message0("CORE_ID: ");
 for(i=0; i<N_VANE; i++)
  {
   Message0("%d  ", CORE_ID[i]);
}
 Message0("\n");
   }
  else
   {
    Message0("CORE_ID: %d\n", CORE_ID[0]);
 }

  Message0("MULTIPLE_GAP: %d\n", MULTIPLE_GAP);
  if(MULTIPLE_GAP)
 {
   Message0("GAP_ID: ");
   for(i=0; i<N_VANE; i++)
    {
     Message0("%d  ", GAP_ID[i]);
  }
   Message0("\n");
   }
  else
   {
     Message0("GAP_ID: %d\n", GAP_ID[0]);
 }

  Message0("INNER_TIME_DEP_PROFILE: %d\n", INNER_TIME_DEP_PROFILE);

  Message0("RPM: %-10.2e VANE_WIDTH: %-10.2e GAP_SIZE: %-10.2e\n", RPM, VANE_WIDTH, GAP_SIZE);

  Message0("INIT_VANE_ANGLE: %-6.2f\n", INIT_VANE_ANGLE);

  Message0("NON_EQUALLY_SPACED_VANE: %d\n", NON_EQUALLY_SPACED_VANE);
  if(NON_EQUALLY_SPACED_VANE)
   {
 total_angle = 0;
 for(i=0; i<N_VANE; i++)
  {
        Message0("ANGLE[%d]=%-5.1f ", i, ANGLE[i]*180/M_PI);
        total_angle += ANGLE[i]*180/M_PI;
  }
```

```
      Message0("\n");
      Message0("total angle: %-6.1f\n", total_angle);
}

  Message0("\n******************** Below is the pump core outer infor ********************\n\n");
  print_one_shape(&pump_core_outer_shape);

  Message0("\n******************** Below is the pump core inner infor ********************\n\n");
  print_one_shape(&pump_core_inner_shape);

  if(GAP_ID[0]>0)
   {
    Message0("\n******************** Below is the pump gap outer infor ********************\n\n");
    print_one_shape(&pump_gap_outer_shape);

    Message0("\n******************** Below is the pump gap inner infor ********************\n\n");
    print_one_shape(&pump_gap_inner_shape);
}
}
/* The input file will be read each time the UDF is loaded */

DEFINE_EXECUTE_ON_LOADING(on_loading, libudf)
{
 int i;
 FILE* fpin;

 Message0("\nON_LOADING udf is executing ...\n");

#if PARALLEL
  if(I_AM_NODE_HOST_P)
#endif

{
 fpin = fopen("input.txt","r");
 if(fpin == NULL)
   Message0("Input file does not exist!");
}

#if PARALLEL
  if(I_AM_NODE_HOST_P)
#endif
   {
 fscanf(fpin, "%d %d %d", &OUTER_SHAPE, &INNER_SHAPE, &N_VANE);

 fscanf(fpin, "%d", &MULTIPLE_CORE);
 if(MULTIPLE_CORE)
  {
    CORE_ID = alloc_1d_int_array(N_VANE);
    for(i=0; i<N_VANE; i++)
     {
      fscanf(fpin, "%d", CORE_ID+i);
  }
  }
 else
  {
       CORE_ID = alloc_1d_int_array(1);
       fscanf(fpin, "%d", CORE_ID+0);
  }
 fscanf(fpin, "%d", &MULTIPLE_GAP);
 if(MULTIPLE_GAP)
  {
   GAP_ID = alloc_1d_int_array(N_VANE);
   for(i=0; i<N_VANE; i++)
    {
     fscanf(fpin, "%d", GAP_ID+i);
  }
```

```
      }
     else
      {
       GAP_ID = alloc_1d_int_array(1);
       fscanf(fpin, "%d", GAP_ID+0);
   }
 fscanf(fpin, "%d", &INNER_TIME_DEP_PROFILE);

 fscanf(fpin, REAL_FMT REAL_FMT REAL_FMT REAL_FMT, &RPM, &R_INNER, &VANE_WIDTH, &GAP_SIZE);
 fscanf(fpin, REAL_FMT REAL_FMT REAL_FMT, &R_OUTER, &DELTA, &INIT_VANE_ANGLE);

 fscanf(fpin, "%d",&NON_EQUALLY_SPACED_VANE);

 if(NON_EQUALLY_SPACED_VANE)
  {
   ANGLE = alloc_1d_real_array(N_VANE);
   for(i=0; i<N_VANE; i++)
    {
     fscanf(fpin, REAL_FMT, ANGLE+i);
     ANGLE[i] *=M_PI/180.0;
  }
     }
 fclose(fpin);
  }
 host_to_node_int_7(OUTER_SHAPE, INNER_SHAPE, N_VANE, MULTIPLE_CORE, MULTIPLE_GAP,
INNER_TIME_DEP_PROFILE, NON_EQUALLY_SPACED_VANE);

 #if RP_NODE
if(MULTIPLE_CORE)
  CORE_ID = alloc_1d_int_array(N_VANE);
else
     CORE_ID = alloc_1d_int_array(1);

if(MULTIPLE_GAP)
  GAP_ID = alloc_1d_int_array(N_VANE);
else
     GAP_ID = alloc_1d_int_array(1);

if(NON_EQUALLY_SPACED_VANE)
    ANGLE = alloc_1d_real_array(N_VANE);
 #endif

if(NON_EQUALLY_SPACED_VANE)
     host_to_node_real(ANGLE, N_VANE);

if(MULTIPLE_CORE)
  host_to_node_int(CORE_ID, N_VANE);
else
  host_to_node_int(CORE_ID, 1);

if(MULTIPLE_GAP)
  host_to_node_int(GAP_ID, N_VANE);
else
  host_to_node_int(GAP_ID, 1);

host_to_node_real_7(RPM, R_INNER, VANE_WIDTH, GAP_SIZE, R_OUTER, DELTA, INIT_VANE_ANGLE);

/* Read in any profile data files */
initialize_shape();
}

DEFINE_EXECUTE_AT_END(acclerate)
{
 RPM *= 1.01;
 Message("\n New RPM is %f \n", RPM);
}
```

EK-F

SHELL SPİRAX S4 ATF HDX

Shell Spirax S4 ATF HDX, sentetik bir üründür.

- Otomotiv sektöründe otomatik şanzımanlarda,
- Otomotiv hidrolik sistemlerinde,
- Hidrolik direksiyonlarda,
- Bazı manuel şanzımanlarda kullanılmaktadır.

Tipik Fiziksel Özellikleri		
Kinematik Viskozite 40 °C 'de mm²/s 100 °C 'de mm²/s	ISO 3108	 33,2 7,2
Viskozite İndeksi	ISO 2909	189
Yoğunluk (15 °C 'de) kg/m³	ISO 12185	847
Parlama Noktası COC °C	ISO 2592	152
Akma Noktası °C	ISO 3016	-48

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mehmet Emin ÇETİN

Doğum Tarihi ve Yeri : 28 Haziran 1988 – Konya

Yabancı Dili : İngilizce

E-posta : cetinmehmetemin@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Yüksek Lisans	Makina Mühendisliği	Yıldız Teknik Üniversitesi	2015
Lisans	Makina Mühendisliği	Yıldız Teknik Üniversitesi	2012
Lise	Fen	Konya Lisesi	2006

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2012	İstanbul Ulaşım San ve Tic. A.Ş.	Makina Mühendisi