

MATLAB Programı Yardımıyla Pompa Tasarımı

Özlem Özenbaş

YÜKSEK LİSANS TEZİ

Makine Mühendisliği Anabilim Dalı

Ocak 2010

Pump Design by Using MATLAB

Özlem Özenbaş

MASTER OF SCIENCE THESIS

Department of Mechanical Engineering

January 2010

MATLAB Programı Yardımıyla Pompa Tasarımı

Özlem Özenbaş

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Makine Mühendisliği Anabilim Dalı
Enerji Termodinamik Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Prof. Dr. Yaşar PANCAR

Ocak 2010

ONAY

Makine Mühendisliđi Anabilim Dalı Yüksek Lisans öđrencisi Özlem Özenbař'ın YÜKSEK LİSANS tezi olarak hazırladıđı "MATLAB Programı Yardımıyla Pompa Tasarımı" bařlıklı bu alıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek kabul edilmiřtir.

Danıřman : Prof. Dr.Yařar PANCAR

İkinci Danıřman : -

Yüksek Lisans Tez Savunma Jürisi:

Üye : Prof. Dr.Yařar PANCAR

Üye : Prof. Dr. Kemal TANER

Üye : Prof. Dr. Nejat KIRA

Üye : Yrd. Do. Dr. İrfan ÜREYEN

Üye : Yrd. Do. Dr. Mesut TEKKALMAZ

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıřtır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Bu çalışmanın ilk bölümünde pompalarla ilgili temel kavramlar ele alınarak pompa tipleri ve hangi pompanın nerede kullanılmasının uygun olacağı araştırıldı. Çalışmada pompaların çalışma prensipleri ve uygun pompayı tasarlayabilmek için kullanılan yöntemler ilgili formüllerle birlikte tek tek incelenerek örnek tasarımlar yapıldı. Bu tasarımların bilgisayar programına (MATLAB) aktarılması için akış diyagramları oluşturularak bilgisayar yardımıyla tasarım gerçekleştirildi ve programlama aşamaları açıklandı. Sonuç olarak tasarımın programa aktarılabilirliği kanıtlanmış oldu.

Anahtar Kelimeler: MATLAB, paletli pompa, dişli pompa, radyal santrifüj pompa, aksenal santrifüj pompa

SUMMARY

In the first part of this study, the basic concepts of the pump has been investigated. Pump types and operation sites have been determined. Then, design principles of pumps and formula have been analysed. Sample designs have been studied. The flow diagrams have been prepared in order to cooperate with computer program (MATLAB). At the end, the transferability of the design program has been shown.

Keywords: MATLAB, vane pump, gear pump, radial centrifugal pump, axial pump

TEŐEKKÜR

Bu tezin hazırlanmasında desteęini esirgemeyen ve görüşleriyle yönlendiren danışmanım Sayın Prof. Dr. Yaşar PANCAR'a teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ	xii
SİMGELER VE KISALTMALAR DİZİNİ	xiii
1. GİRİŞ	1
2. PALETLİ POMPA	2
2.1. Paletli Pompa Türleri.....	3
2.1.1. Tek odalı paletli pompalar	3
2.1.2. Çift odalı paletli pompalar	4
2.2. Paletli Pompa Ana Elemanları.....	5
2.3. Paletli Pompa Tasarımında Kullanılan Formüller	6
2.4. Paletli Pompa Tasarım Programı Akış Şeması	9
2.4.1. Ana program.....	9
2.4.2. Alt program	10
2.5. Paletli Pompa Tasarım Programının Uygulanması.....	12
2.5.1. Paletli pompa ile ilgili hesaplamalar	12
2.5.2. Paletli pompa tasarım yazılımının oluşturulması	14
3. DIŞLI POMPA	20
3.1. Dişli Pompa Türleri	20
3.1.1. Dıştan dişli pompalar.....	20
3.1.2. İçten dişli pompalar	21
3.1.3. Gerotor pompalar	22
3.1.4. Vidalı pompalar.....	22
3.2. Dişli Pompa Ana Elemanları	23
3.3. Dişli Pompa Tasarımında Kullanılan Formüller.....	24
3.4. Dişli Pompa Tasarım Programı Akış Şeması	28
3.4.1. Ana program.....	28
3.4.2. Alt program	29

İÇİNDEKİLER(devam)

	<u>Sayfa</u>
3.5. Dişli Pompa Tasarım Programının Uygulanması	31
3.5.1. Dişli pompa ile ilgili hesaplamalar	31
3.5.2. Dişli pompa tasarım yazılımının oluşturulması	33
4. RADYAL SANTRİFÜJ POMPA.....	39
4.1. Radyal Santrifüj Pompa Ana Elemanları.....	40
4.1.1. Salyangoz	40
4.1.2. Çark	40
4.1.3. Difüzör (Yayıcı)	41
4.1.4. Pompa Mili	41
4.2. Radyal Santrifüj Pompa Tasarımında Kullanılan Formüller	42
4.3. Radyal Santrifüj Pompa Tasarım Programı Akış Şeması	56
4.3.1. Ana program.....	56
4.3.2. Alt program	59
4.4. Radyal Santrifüj Pompa Tasarım Programının Uygulanması.....	64
4.4.1. Radyal santrifüj pompa ile ilgili hesaplamalar	64
4.4.2. Radyal santrifüj pompa tasarım yazılımının oluşturulması	73
5. EKSENEL SANTRİFÜJ POMPA.....	79
5.1. Eksenel Santrifüj Pompa Ana Elemanları	80
5.2. Eksenel Santrifüj Pompa Tasarımında Kullanılan Formüller	80
5.3. Eksenel Santrifüj Pompa Tasarım Programı Akış Şeması.....	86
5.3.1. Ana program.....	86
5.3.2. Alt program	89
5.4. Eksenel Santrifüj Pompa Tasarım Programının Uygulanması	91
5.4.1. Eksenel santrifüj pompa ile ilgili hesaplamalar.....	91
5.4.2. Eksenel santrifüj pompa tasarım yazılımının oluşturulması.....	94
6. SONUÇ VE ÖNERİLER.....	101
7. KAYNAKLAR DİZİNİ	103

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.3. Paletli pompa ana elemanları	5
2.4. Paletli pompa montajının gösterilmesi.....	5
2.5. Paletli pompaya ait ana programın akış şeması	9
2.6. Paletli pompaya ait alt programın akış şeması	11
2.7. “Guide Quick Start” ekranı	14
2.8. Paletli pompa GUI ön görünüm ekranı	15
2.9. GUI uicontrol ekranı	15
2.10. GUI elemanlarında isimlendirme	16
2.11. Paletli pompa hesap ve çizim ekranı	18
2.12. Paletli pompa ekranında “HESAP” butonuna basılmış durum	18
2.13. Paletli pompa ekranında “ÇİZİM” butonuna basılmış durum	19
3.1. Dıştan dişli pompa (MEGEP, 2007)	21
3.2. İçten dişli pompa (MEGEP, 2007).....	21
3.3. Gerotor pompa (http://extreme.infomagic.net/static/SmGerotor.jpeg)	22
3.4. Vidalı pompa (MEGEP, 2007).....	23
3.5. Dişli pompa ana elemanları.....	23
3.6. Dişli pompanın gösterimi	24
3.7. Dişli pompaya ait ana programın akış şeması.....	28
3.8. Dişli pompaya ait alt programın akış şeması	30
3.9. “Guide Quick Start” ekranı	33
3.10. Dişli pompa GUI ön görünüm ekranı.....	34
3.11. GUI uicontrol ekranı	34
3.12. GUI elemanlarında isimlendirme	35
3.13. Dişli pompa hesap ve çizim ekranı	37
3.14. Dişli pompa ekranında “HESAP” butonuna basılmış durum.....	37
3.15. Dişli pompa ekranında “ÇİZİM” butonuna basılmış durum	38
4.1. Santrifüj pompa görünüşü	39
4.2. Salyangozun görünüşü	40
4.3. Çarkın görünüşü	41
4.4. Mil ve gövde arası bağlantının görünüşü	41
4.5. Mil ve motor bağlantısı	42
4.6. Özgül hız-pompa verim diyagramı (Pancar, 2007).....	43
4.7. K_{cm1} ve K_{cm2} diyagramı (Pancar, 2007).....	45

ŞEKİLLER DİZİNİ(devam)

<u>Şekil</u>	<u>Sayfa</u>
4.8. Giriş hız üçgeni	47
4.9. Basınç katsayısı ve D_1/D_2 oranları (Pancar, 2007).....	49
4.10. Çıkış hız üçgeni.....	50
4.11. $K_v - n_s$ ve $(D_3-D_2) / D_2 - n_s$ ilişkisi (Pancar, 2007)	53
4.12. Salyangoz	54
4.13. Meridyen kesiti ve kanatlar	55
4.14. Radyal santrifüj pompaya ait ana programın akış şeması	58
4.15. Radyal santrifüj pompaya ait alt programın akış şeması.....	63
4.16. “Guide Quick Start” ekranı	73
4.17. Radyal santrifüj pompa GUI ön görünüm ekranı	74
4.18. GUI uicontrol ekranı	74
4.19. GUI elemanlarında isimlendirme	75
4.20. Radyal santrifüj pompa hesap ve çizim ekranı.....	77
4.21. Radyal santrifüj pompa ekranında “HESAP” butonuna basılmış durum.....	77
4.22. Radyal santrifüj pompa ekranında “ÇİZİM” butonuna basılmış durum	78
4.1. Eksenel santrifüj pompa (Özgür, 1972)	79
5.2. Eksenel santrifüj pompa elemanları (Özgür, 1972)	80
5.3. n_s - ku - dm/D diyagramı (Özgür, 1972).....	81
5.4. Eksenel santrifüj pompa kanat ana ölçüleri (Özgür,1972).....	84
5.5. Eksenel santrifüj pompaya ait ana programın akış şeması.....	88
5.6. Eksenel santrifüj pompaya ait alt programın akış şeması	90
5.7. “Guide Quick Start” ekranı	95
5.8. Eksenel santrifüj pompa GUI ön görünüm ekranı	95
5.9. GUI uicontrol ekranı	96
5.10. GUI elemanlarında isimlendirme	96
5.11. Eksenel santrifüj pompa hesap ve çizim ekranı	99
5.12. Eksenel santrifüj pompa ekranında “HESAP” butonuna basılmış durum	99
5.13. Eksenel santrifüj pompa ekranında “ÇİZİM” butonuna basılmış durum.....	100

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1. Düzeltilmiş daire açınımlı profil diş yaylarının çiziminde kullanılan r_1 ve r_2 yarıçaplarının değerini gösteren çizelge	26
3.2. Modül (M) Standartları (DIN 780)	26
4.1. Emniyet Katsayısı	44
4.2. Nokta Nokta Metodu için hazırlanan çizelge	55
5.1. Aerodinamik profillerin ölçüleri (Troslanski, 1982)	85
5.2. Kanat çiziminde kullanılan formüller	93

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
C_{0m}	Akışkanın çarka girmeden önceki hızı (m/s)
r_2	Alt yay (mm)
P	Basınç (bar)
H	Basma yüksekliği (m)
b_1	Çark giriş genişliği (mm)
β_1	Çark giriş kanat açısı ($^\circ$)
β_{1d}	Çark giriş kanat dış açısı ($^\circ$)
D_{1d}	Çark giriş kanat dış çapı (mm)
U_{1d}	Çark giriş kanat dış çevresel hızı (m/s)
β_{1i}	Çark giriş kanat iç açısı ($^\circ$)
D_{1i}	Çark giriş kanat iç çapı (mm)
U_{1i}	Çark giriş kanat iç çevresel hızı (m/s)
U_1	Çark girişindeki akışkan çevresel hızı (m/s)
D_2	Çark çıkış çapı (mm)
U_2	Çark çıkışı akışkan çevresel hızı (m/s)
d_0	Çarka girişteki toplam kesit alanının dış çapı (mm)
λ_2	Çıkış daralma faktörü
Q	Debi (m^3/s)
R	Dış çap (mm)
s_0	Diş genişliği (mm)
t	Diş hatvesi (mm)
z	Diş sayısı (<i>Adet</i>)
d_a	Diş üstü dairesi çapı (mm)
h	Diş yüksekliği (mm)
θ	Dişler arası açısı ($^\circ$)
t	Et kalınlığı (cm)

A_0	Giriş kesit alanı (m^2)
$d_{göbek}$	Göbek çapı (mm)
A_g	Göbek kesit alanı (m^2)
V_B	Gövde hacmi (m^3)
$\eta_{hidrolik}$	Hidrolik verim (%)
Q'	Kaçak debi (m^3)
D_1	Kanat üzerindeki orta akım iplikçiğın çapı (mm)
Z	Kanat sayısı (<i>adet</i>)
σ_t	Malzeme gerilme (kgf/cm^2)
d_{mil}	Mil çapı (mm)
P_{mil}	Mil gücü (<i>BG</i>)
m	Modül
Ne	Motor gücü (<i>BG</i>)
γ	Özgöl ağırlık (kgf/m^3)
n_s	Özgöl hız (d/d)
Dr	Rotor çapı (mm)
V_r	Rotor hacmi (m^3)
C_{m1}	Suyun kanata giriş hızı (m/s)
d_f	Taban dairesi çapı (mm)
d_t	Taksimat dairesi çapı (mm)
n_q	Tanım sayısı
V_a	Tek dişle taşınan hacim (m^3)
d_g	Temel dairesi çapı (mm)
V_g	Tüm dişlerle taşınan hacim (m^3)
r_1	Üst yay (mm)
$\eta_{volumetrik}$	Volümetrik verim (%)

BÖLÜM 1

GİRİŞ

Sıvılara hidrolik enerji kazandırarak taşınmasını sağlayan makinalara pompa denilir (Yalçın, 1998). Su yağ gibi bir akışkanı istenilen basınç ve debide sisteme gönderirler. Bir tür turbo makinedir. Hidrolik pompa dönme hareketini genel olarak sabit sistemlerde elektrik motorunun; taşınabilen sistemlerde içten yanmalı motorun tahriki ile sağlarlar.

Pompalar emiş hattından emdikleri akışkanı basma hattından hacimsel bir küçülme ile basarlar. Bu sayede akışkan düşey ve yatay olarak uzağa taşınmış olur. Sabit debili ve değişken debili tipleri mevcuttur.

Endüstrinin hemen hemen her alanında pompa kullanılmaktadır. İnsan kalbi bile pompaya bir örnektir. Kullanım alanına göre farklı tipte birçok pompa mevcuttur. Bu pompalar da kendi aralarında basınç, debi, güç yönlerinden farklılık gösterebilmektedir.

Pompalara olan ihtiyaç, teknolojinin de gelişmesiyle birlikte her geçen gün artmaktadır. Pompaların; araçlarda benzinin basılması ve yağ iletimi, kazanlarda kimyasal dozlama, soğutma kulelerinde su sirkülasyonu, şebeke suyu iletimi gibi kullanım alanları vardır.

Makine mühendisi, en kısa sürede en az hatayla sistemlerin tasarımını yapar. Bunu yaparken sistemin kullanışlı ve düşük maliyetli olmasını sağlar. Kullanım alanı bu kadar geniş olan pompalar için de bu kural geçerlidir. Pompa tasarımı yaparken bu konuda makine mühendisine en büyük yardımcı; paket programlara bağlı kalmayan makine mühendisinin kendisinin hazırlamış olduğu bilgisayar programıdır. Yapılan bu program ile optimum zaman-iş-maliyet şartları sağlanabilmektedir.

BÖLÜM 2

PALETLİ POMPA

Bir hazne ve bu haznenin içine eksantrik yerleştirilmiş silindirik bir rotordan oluşur. Debi eksantrikliğin artmasına bağlı olarak artar. Paletler, rotor üzerinde açılmış kanallara yerleştirilir ve bu kanallar içinde rotorun dönüşüyle oluşan merkezkaç kuvvetin etkisiyle aşağı yukarı hareket ederler. Ayrıca paletlerin diplerine konmuş olan yaylar paletleri çevreye doğru iterler. İtilen paletler hazneyle rotor arasında kalır. Hazne içinde rotorun dönmesiyle açılan kanatlar vasıtasıyla emilen sıvı basma hattına taşınır. Kanadın itilmesi kanat uçlarında iyi bir sızdırmazlık sağlasa da zamanla kanat ve gövde yüzeyi aşınacaktır. Bunu önlemek için kanatlar kanallı veya kesik uçlu olarak yapılarak kanadın gövdeye aşırı yüklenmesi engellenir. Bu tip pompalarda basma yüksekliği arttıkça palet adedini de artırmak faydalı olur. Belli bir basınç seviyesinin üstünde ve akışkanın yağlama karakteriğine bağlı olarak stator ile paletler arasındaki yağ filmi yırtılabilir. Bu da aşınmaya neden olur. Takriben 150 bar'ın üstündeki sistem basınçlarında çalışan paletli pompalar, bu baskı kuvvetini azaltmak için çift paletli olarak tasarlanır (Karmel, A.M.).

Paletli pompaların yararları aşağıdaki şekilde özetlenebilir;

- Yüksek basınçlarda çalışabilir,
- Kısa süreli kuru çalışma mümkündür,
- Vakum kapasiteleri yüksektir,
- Solventler ve LPG için kullanılabilirler,
- Palet açılmasından kaynaklanabilecek aşınmaları en aza indirir,
- Boyutları küçüktür ve
- Gürültüsüz çalışırlar.

Paletli pompaların mahsurları ise şu şekilde özetlenebilir;

- Yüksek basınca dayanıklı değildirler,
- Yüksek viskoziteli sıvılarda çalışmaları oldukça güçtür,
- Aşındırıcılar ile kullanılamazlar,
- Fazla elemana sahiptirler,
- İşçilikleri fazladır ve
- Kaçaklar fazladır.

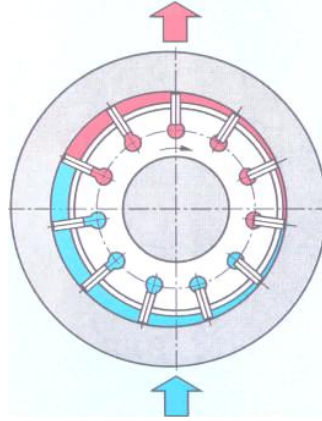
2.1. Paletli Pompa Türleri

- Tek odalı paletli pompalar
- Çift odalı paletli pompalar

İki tipte de palet ve rotor bulunur. Aradaki fark, hazne şeklinden kaynaklıdır.

2.1.1. Tek odalı paletli pompalar

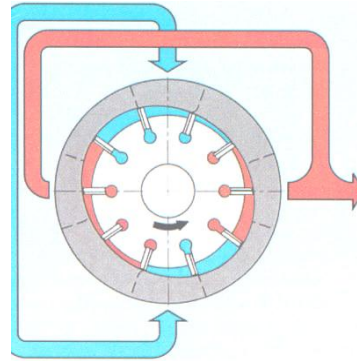
Paletlerin strok hareketi dairesel içyapıya sahip bir halka (stator) ile sınırlanır. Rotor eksenine göre eksantrik yerleştirilmiş olan halka nedeniyle, deplasman odacıklarının hacmi devamlı olarak değiştirilir. Odacıkların doldurulması (emiş) ve boşaltılması işlemi, çift odalı paletli pompalardakinin aynısıdır (Exner, et al., 1998).



Şekil 2.1. Tek odalı paletli pompa (MEGEP, 2007)

2.1.2. Çift odalı paletli pompalar

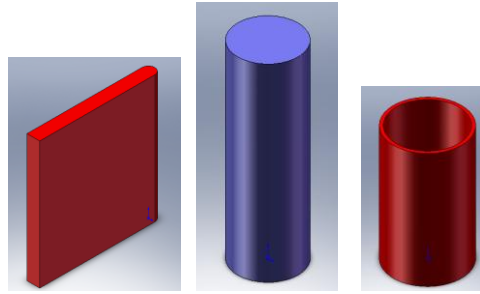
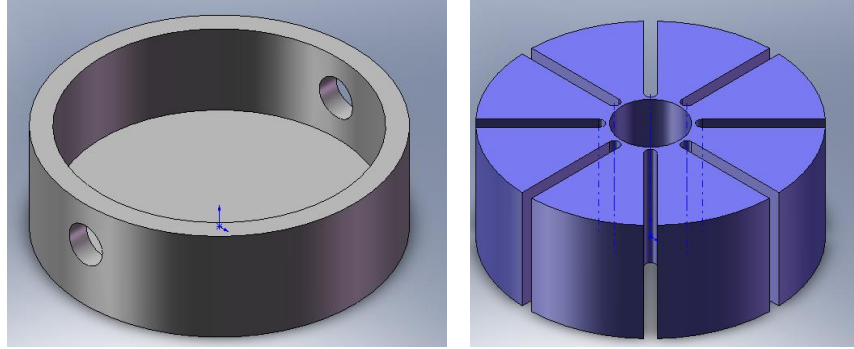
Stator, çift kam oluşturan bir iç yüzeye sahiptir. Bu, her bir paletin, milin her bir dönüşünde iki kez strok yapmasına neden olur. İletim odacıkları; iki palet, statorun iç yüzeyi ve yanlardaki kontrol plakaları arasında oluşturulur.



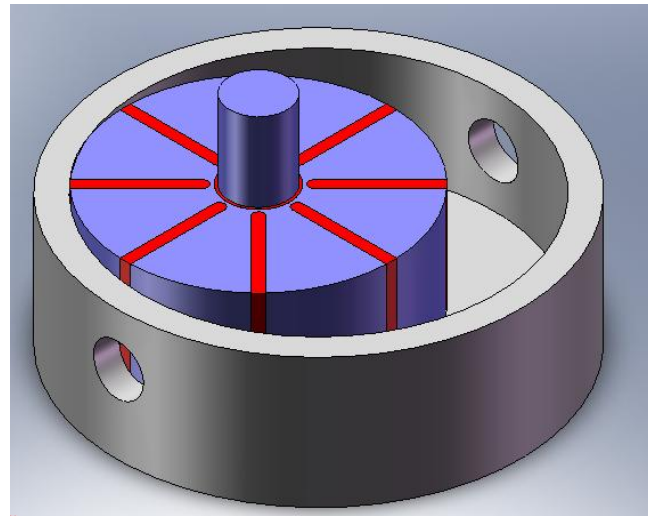
Şekil 2.2. Çift odalı paletli pompa (MEGEP, 2007)

2.2. Paletli Pompa Ana Elemanları

Bir paletli pompanın ana elemanları gövde, rotor, paletler, pompa mili, mil yatakları ve sistemi dış etkilerden koruyan bir haznedan oluşur.



Şekil 2.3. Paletli pompa ana elemanları



Şekil 2.4. Paletli pompa montajının gösterilmesi

2.3. Paletli Pompa Tasarımında Kullanılan Formüller

Paletli pompa tasarımında aşağıdaki donelerin bilinmesi gereklidir:

- Basınç
- Dış çap
- Devir sayısı
- Palet sayısı
- Palet yüksekliği
- Malzeme gerilme
- Kullanılacak akışkanın özgül ağırlığı

Et kalınlığının hesaplanması;

Verilen dış çapa göre et kalınlığı ve emme-basma boru çapları hesaplanır. Hesaplanan değerlere ve yüksekliğe göre hazne çizilir.

$$t = \frac{P \cdot R}{2 \cdot \sigma_t} \quad (2.1)$$

Emme basma boru çaplarının hesaplanması;

$$R \leq 100mm \rightarrow \text{Emme Basma Boru Çapı} = 1.2''$$

$$R > 100mm \rightarrow \text{Emme Basma Boru Çapı} = 3/4''$$

Basma yüksekliğinin hesaplanması;

$$H = \frac{P}{\gamma} \quad (2.2)$$

Rotor apının hesaplanması (Yaklaşık hesap);

Rotor apı ve palet boyu yaklaşık olarak hesaplanır. Palet kalınlığı ve boyu da göz önünde bulundurularak rotor ve paletler çizilir. Mil ve göbek apı hesaplanarak rotor merkezine göbek apı ölçüsünde delik açılır.

$$Dr = 0.75 \cdot R \quad (2.3)$$

Palet boyunun hesaplanması (Yaklaşık hesap);

$$l = R - Dr + 3 \quad (2.4)$$

Bütün hacmin hesaplanması;

$$V_B = \frac{\pi \cdot R^2}{4} \cdot b \quad (2.5)$$

Rotor hacminin hesaplanması;

$$V_r = \frac{\pi \cdot Dr^2}{4} \cdot b \quad (2.6)$$

Debinin hesaplanması;

$$Q = \frac{(V_B - V_r) \cdot n}{60} \quad (2.7)$$

Gücün hesaplanması;

$$Ne = \frac{\gamma \cdot Q \cdot Hm}{75 \cdot \eta_g} \quad BG \quad (2.8)$$

Mil apının hesaplanması;

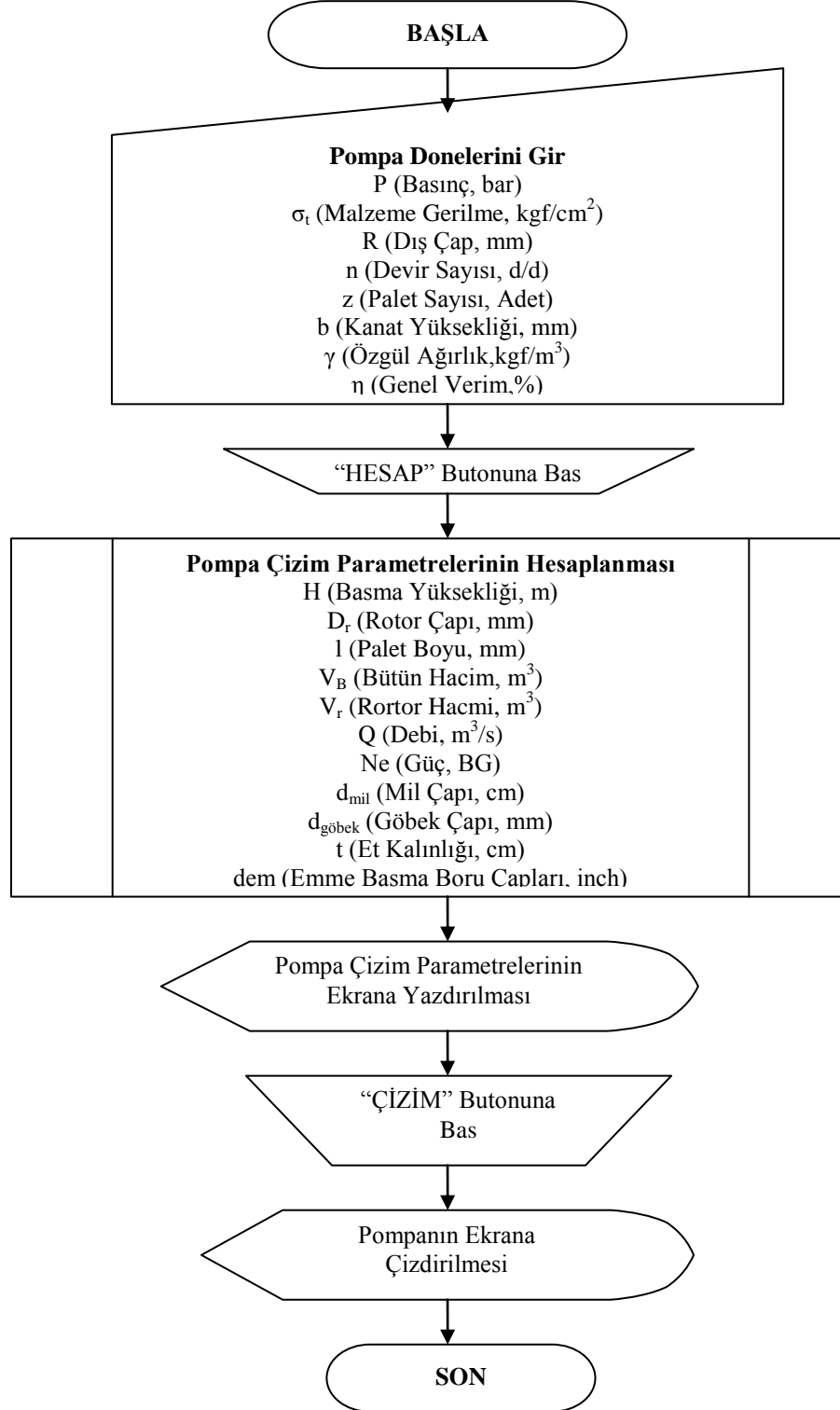
$$d_{mil} = 14.4 \cdot \sqrt[3]{\left(\frac{Ne}{n}\right)} \quad (2.9)$$

Göbek apının hesaplanması (yaklaşık hesap);

$$d_{göbek} = d_{mil} + 3mm \quad (2.10)$$

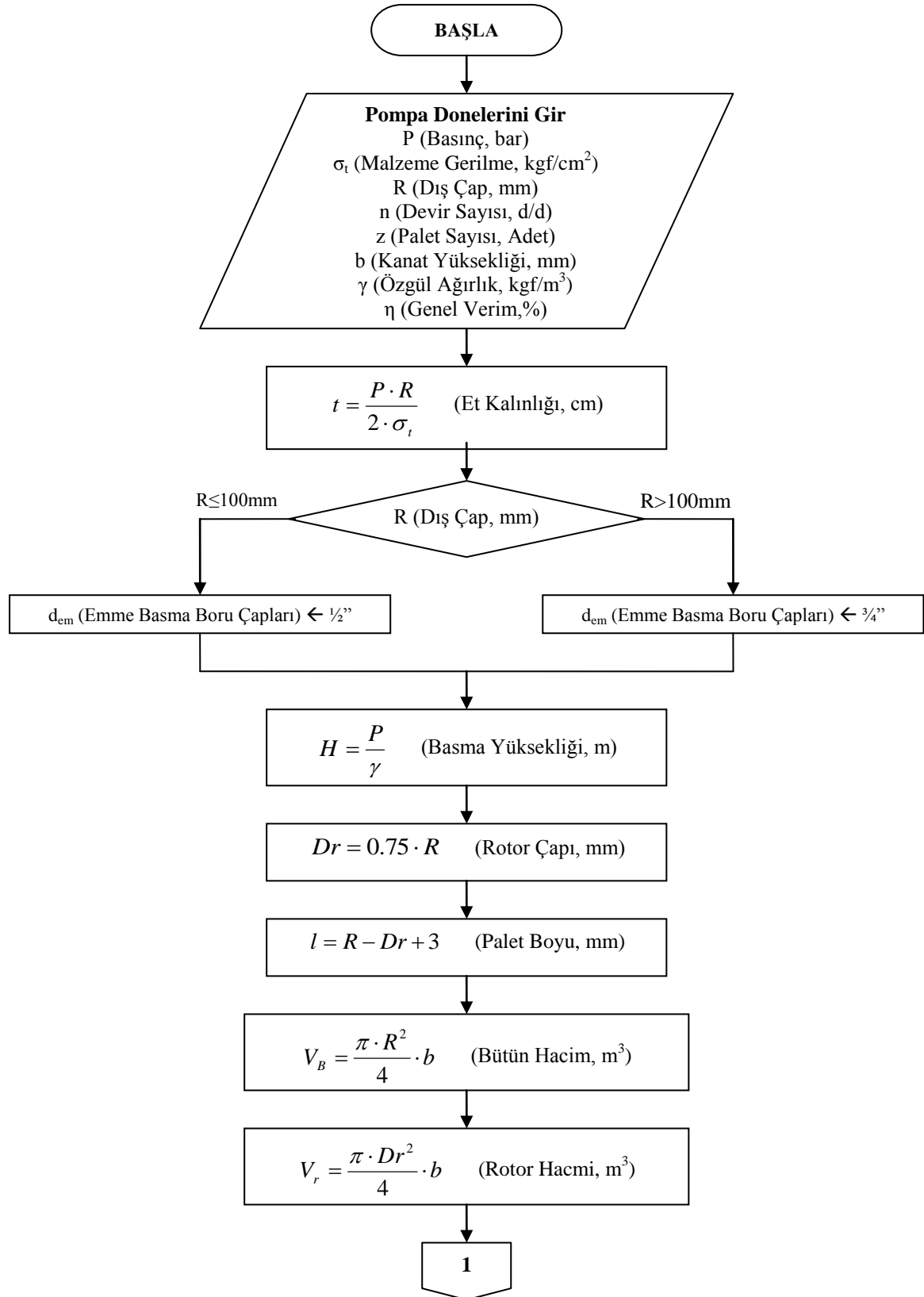
2.4. Paletli Pompa Tasarım Programı Akış Şeması

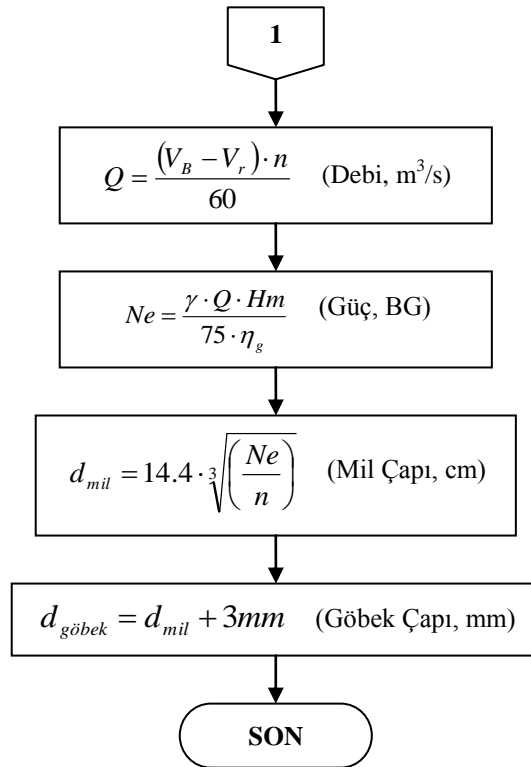
2.4.1. Ana program



Şekil 2.5. Paletli pompaya ait ana programın akış şeması

2.4.2. Alt program





Şekil 2.6. Paletli pompaya ait alt programın akış şeması

2.5. Paletli Pompa Tasarım Programının Uygulanması

Pompa doneleri:

$$P = 60bar \text{ (Basınç)}$$

$$R = 135mm \text{ (Dış Çap)}$$

$$n = 1000d / d \text{ (Devir Sayısı)}$$

$$z = 8adet \text{ (Palet Sayısı)}$$

$$b = 40mm \text{ (Kanat Yüksekliği)}$$

$$\sigma_t = 420kgf / cm^2 \text{ (Malzeme Gerilme)}$$

$$\gamma = 850kgf / m^3 \text{ (Özgül Ağırlık)}$$

2.5.1. Paletli pompa ile ilgili hesaplamalar

Et kalınlığının hesaplanması;

$$t = \frac{P \cdot R}{2 \cdot \sigma_t} = \frac{(60bar) \cdot (135mm)}{2 \cdot (420kgf / cm^2)} = 0.96cm$$

Emme basma boru çaplarının hesaplanması;

$$R > 100mm \rightarrow \text{Emme Basma Boru Çapı} = 3 / 4''$$

Basma yüksekliğinin hesaplanması;

$$H = \frac{P}{\gamma} = \frac{(60000kgf / m)}{(850kgf / m^3)} = 70.5852m$$

Rotor çapının hesaplanması;

$$Dr = 0.75 \cdot R = 0.75 \cdot (135mm) = 102.6mm$$

Palet boyunun hesaplanması;

$$l = R - Dr + 3 = (135mm) - (102.6mm) + 3 = 35.4mm$$

Bütün hacmin hesaplanması;

$$V_B = \frac{\pi \cdot R^2}{4} \cdot b = \frac{\pi \cdot (0.135m)^2}{4} \cdot (0.04m) = 0.00057256 m^3$$

Rotor hacminin hesaplanması;

$$V_r = \frac{\pi \cdot Dr^2}{4} \cdot b = \frac{\pi \cdot (0.102.6m)^2}{4} \cdot (0.04m) = 0.00033071 m^3$$

Debinin hesaplanması;

$$Q = \frac{(V_B - V_r) \cdot n}{60} = \frac{(0.00024185 m^3) \cdot 1000}{60} = 0.0040308 m^3 / s$$

Gücün hesaplanması;

$$Ne = \frac{\gamma \cdot Q \cdot Hm}{75 \cdot \eta_g} = \frac{(850kgf / m^3) \cdot (0.0040308 m^3 / s) \cdot (70.5882m)}{75 \cdot 0.95} = 3.3943 BG$$

Standart güç olarak 4BG alınır.

Mil çapının hesaplanması;

$$d_{mil} = 14.4 \cdot \sqrt[3]{\left(\frac{Ne}{n}\right)} = 14.4 \cdot \sqrt[3]{\left(\frac{4BG}{1000 d / d}\right)} = 2.28cm$$

Standart mil çapı olarak $20mm$ alınır.

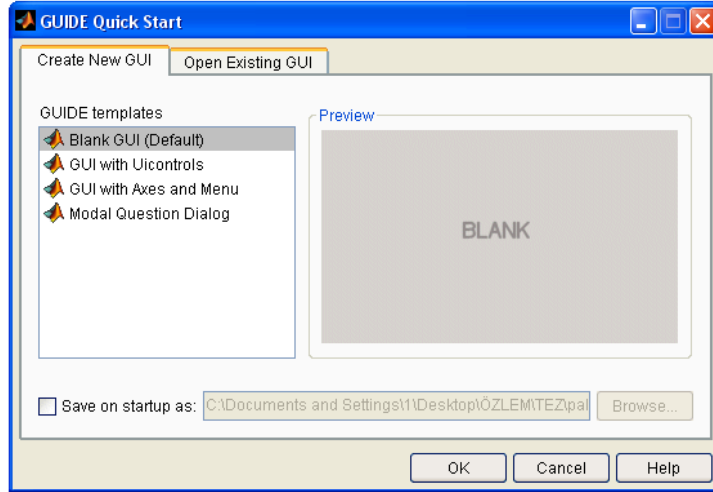
Göbek çapının hesaplanması;

$$d_{göbek} = d_{mil} + 3mm = (20mm) + (3mm) = 23mm$$

2.5.2. Paletli pompa tasarım yazılımının oluşturulması

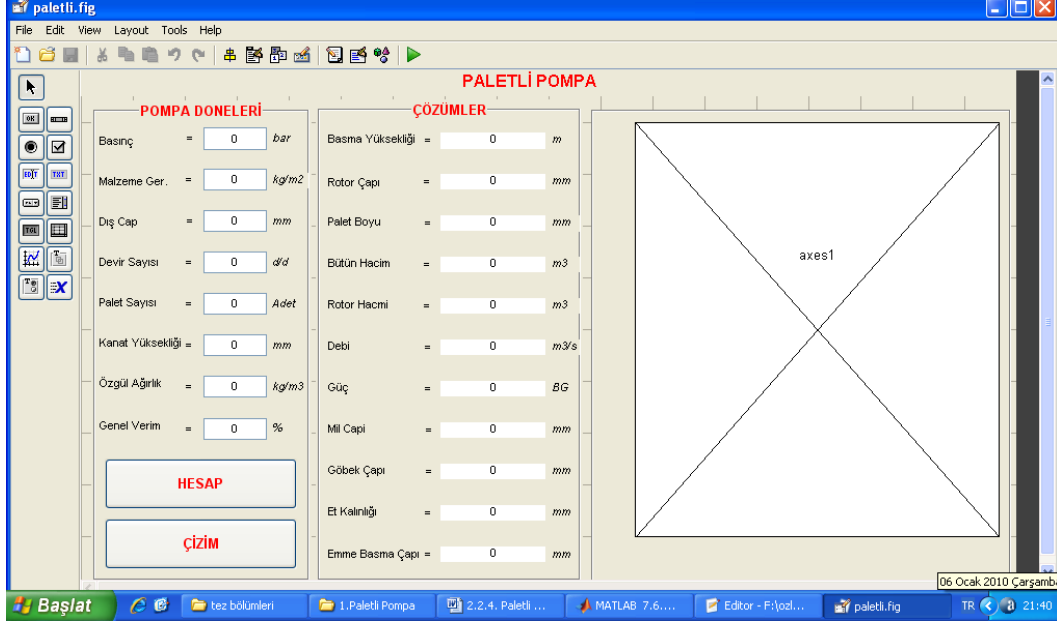
GUI'nin görüntülü kısmını oluşturmak:

- Komut Satırına “>>guide” yazılır.
- Gelen ekrandan Blank GUI ikonunu işaretler ve OK'ye basılır.



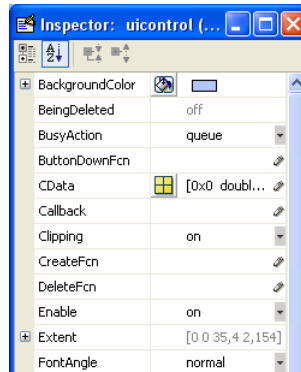
Şekil 2.7. “Guide Quick Start” ekranı

- GUI'nin nasıl görüneceğine karar verilir. Şöyle görünebilir:



Şekil 2.8. Paletli pompa GUI ön görünüm ekranı

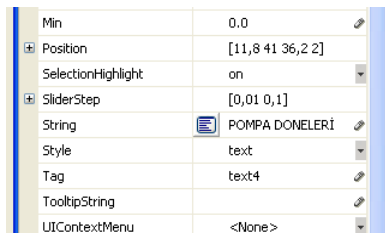
- Sol taraftaki Menüden StaticText, EditText, Axes ve PushButton'lar sürüklenip ekrana bırakılır. Yerlerine yerleştirilir.
- StaticText'lerden biri seçilip çift tıklanır. Aşağıdaki gibi bir ekranla karşılaşılır:



Şekil 2.9. GUI uicontrol ekranı

String kısmı “POMPA DONELERİ”ne dönüştürülür. İsternirse yazı boyutu, yazı kalınlığı gibi değerler de bu kısımdan değiştirilebilir.

Aynı işlem diğer StaticTextlere de uygulanır. String kısımları Şekil 2.10’daki gibi Basinc, =, 0, bar gibi değerlerle değiştirilir.



Şekil 2.10. GUI elemanlarında isimlendirme

- Çözümler bölümündeki sonuçların görüneceği StaticTextler’in String kısımlarına “0”, Tag kısımlarına da sırasıyla şu değerler yazılır:

answer_basmayuksekligi_staticText
 answer_rotorcapi_staticText
 answer_paletboyu_staticText
 answer_butunhacim_staticText
 answer_rotorhacmi_staticText
 answer_debi_staticText
 answer_motorgucu_staticText
 answer_milcapi_staticText
 answer_gobekcapi_staticText
 answer_etkalinligi_staticText
 answer_emmebasma_staticText

- EditText’lerden birine çift tıklanıp String kısmına “0”, Tag kısmına basinc_editText yazılır.

- Diğer EditText'lerin String kısımlarına da "0" yazılır ancak Tag kısımlarına sırasıyla şu değerleri yazılır:

malzemegerilme_editText

discap_editText

devirsayisi_editText

paletsayisi_editText

kanatyuksekligi_editText

ozgulagirlik_editText

genelverim_editText

- PushButton'lardan birine çift tıklanıp String kısmına "HESAP", Tag kısmına hesap_pushbutton yazılır.

- Diğer PushButton'un String kısımlarına da "CIZIM", Tag kısmına anaelemancizim_pushbutton yazılır.

- Axes'e çift tıklanıp Tag kısmına axes1 yazılır.

- Sayfa "paletli" olarak kaydedilir. paletli.fig ve paletli.m olmak üzere iki dosya oluşur. Oluşan m.file'ı açarız.

GUI'ye kod yazmak:

paletli.m isimli m.file dosyasına Ekler'de verilen kodlar yazılır.

Run  ikonuna tıklandığında Şekil 2.11'deki görünüm elde edilir:

POMPA DONELERİ		ÇÖZÜMLER	
Basınç	= 0 bar	Basma Yüksekliği	= 0 m
Malzeme Ger.	= 0 kg/m ²	Rotor Çapı	= 0 mm
Dış Cap	= 0 mm	Palet Boyu	= 0 mm
Devir Sayısı	= 0 d/d	Bütün Hacim	= 0 m ³
Palet Sayısı	= 0 Adet	Rotor Hacmi	= 0 m ³
Kanat Yüksekliği	= 0 mm	Debi	= 0 m ³ /s
Özgül Ağırlık	= 0 kg/m ³	Güç	= 0 BG
Genel Verim	= 0 %	Mil Capi	= 0 mm
<input type="button" value="HESAP"/> <input type="button" value="ÇİZİM"/>		Göbek Çapı	= 0 mm
		Et Kalınlığı	= 0 mm
		Emme Basma Çapı	= 0 mm

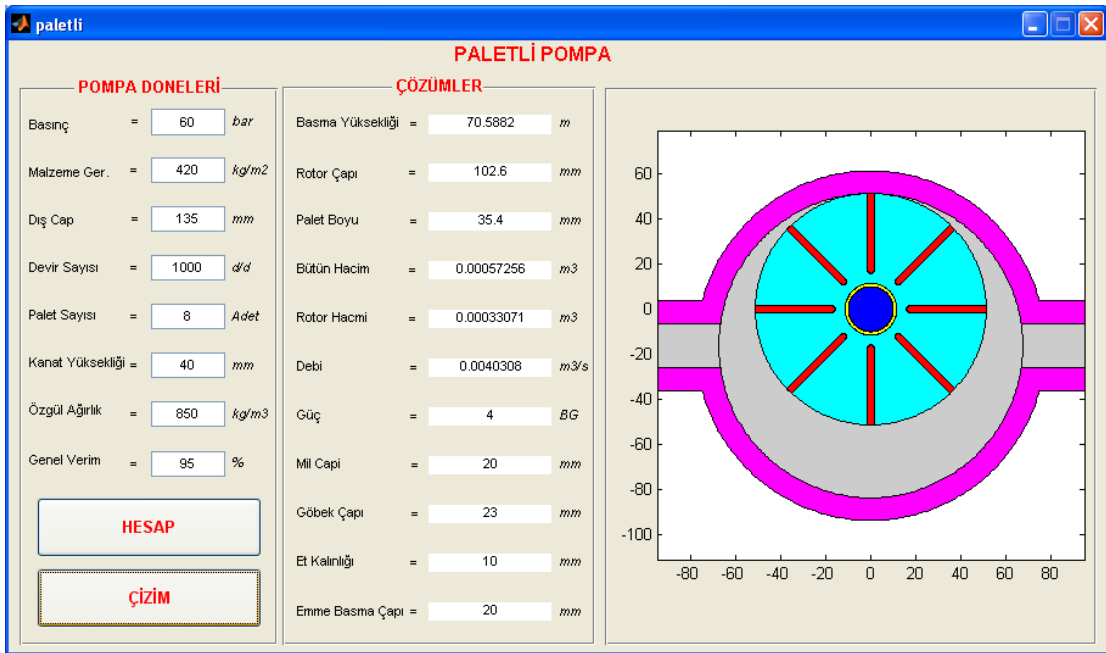
Şekil 2.11. Paletli pompa hesap ve çizim ekranı

Sol taraftaki pompa donelerini girerek “HESAP” butonuna bastığımızda MATLAB hesapları yapar ve sonuçlar ekranda görüntülenir.

POMPA DONELERİ		ÇÖZÜMLER	
Basınç	= 60 bar	Basma Yüksekliği	= 70.5882 m
Malzeme Ger.	= 420 kg/m ²	Rotor Çapı	= 102.6 mm
Dış Cap	= 135 mm	Palet Boyu	= 35.4 mm
Devir Sayısı	= 1000 d/d	Bütün Hacim	= 0.00057256 m ³
Palet Sayısı	= 8 Adet	Rotor Hacmi	= 0.00033071 m ³
Kanat Yüksekliği	= 40 mm	Debi	= 0.0040308 m ³ /s
Özgül Ağırlık	= 850 kg/m ³	Güç	= 4 BG
Genel Verim	= 95 %	Mil Capi	= 20 mm
<input type="button" value="HESAP"/> <input type="button" value="ÇİZİM"/>		Göbek Çapı	= 23 mm
		Et Kalınlığı	= 10 mm
		Emme Basma Çapı	= 20 mm

Şekil 2.12. Paletli pompa ekranında “HESAP” butonuna basılmış durum

“ÇİZİM” butonuna basıldığında ana elemanların çizimi axes bölümünde görüntülenir.



Şekil 2.13. Paletli pompa ekranında “ÇİZİM” butonuna basılmış durum

Program MATLAB kurulu olmayan bilgisayarda bu haliyle çalışmaz. Çalışmasını sağlamak için programı .exe uzantılı hale dönüştürmemiz gerekir. Bunun için m.file açıkken komut penceresine şu komutu yazarız:

```
>>mcc – m paletli
```

Lcc compiler seçeneğini seçtiğimizde program .exe uzantılı hale gelir.

BÖLÜM 3

DİŞLİ POMPA

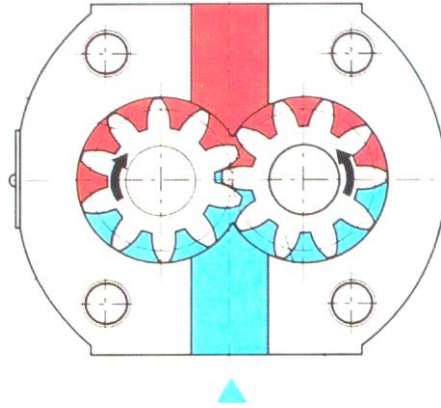
Bir dişli pompa iki alın dişlisinden oluşur. Bu dişlilerden biri motorla tahrik edilince diğeri de zıt yönde döner ve sıvıyı emme kanalından basma kanalına doğru iter. Diş boşluklarına alınan sıvı sisteme gönderilerek pompalama işlemi gerçekleştirilir. Debileri sabittir. Doğrudan elektrik motoruna bağlanarak çalıştırılırlar. Devir sayısı 1000–1500 arasında tutulur. Daha yüksek devirlerde titreşim artar. Diş sayısının artması basılan sıvıdaki düzgünsüzlükleri azaltır.

3.1. Dişli Pompa Türleri

3.1.1. Dıştan dişli pompalar

En çok kullanılan dişli pompa tipidir. Hafif olmalarına rağmen yüksek basınç üretebilmektedir. Bu da büyük bir avantaj sağlamaktadır. Maliyeti düşük, tasarımı basittir. Devir sayısı aralığı geniştir. Değişik viskozitedeki sıvıları taşıyabilirler. Aynı çapta iki dişliden oluşabildiği gibi farklı çaptaki dişlilerden de oluşabilir. Dişli çapları farklı sistemlerde her bir dişin devir sayısı da farklı olur.

Dişli pompalar genel olarak yağ pompası olarak kullanılırlar. Böylece çalışma sırasında gerekli yağlama işi de kendiliğinden sağlanmış olur.

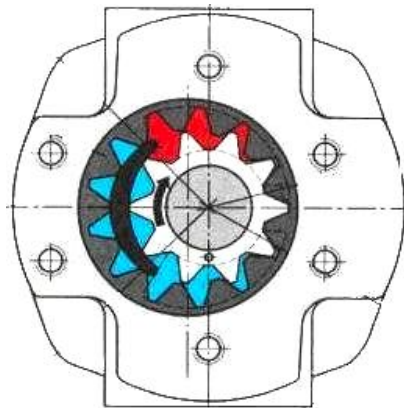


Şekil 3.1. Dıştan dişli pompa (MEGEP, 2007)

3.1.2. İçten dişli pompalar

Yuvarlanma daireleri içten teğettir. Dış dişli iç dişlinin döndüğü yönde serbestçe döner. Gövde zemini üzerindeki hilal bu iki dişli arasında sızdırmazlık elemanı olarak görev yapar. Çevirici dişli, emiş ağzında vakum yaratarak dişler arasına sıvıyı alır. Hilalle iç dişli arasında kalan sıvı basma tarafında dış dişli tarafından basılır.

İçten dişli pompalar, dıştan dişli pompalara göre daha az gürültü yaratırlar. Pres, plastik makineleri gibi endüstriyel hidrolik sistemlerde ve kapalı ortamlarda çalışan araçlarda kullanılırlar.



Şekil 3.2. İçten dişli pompa (MEGEP, 2007)

3.1.3. Gerotor pompalar

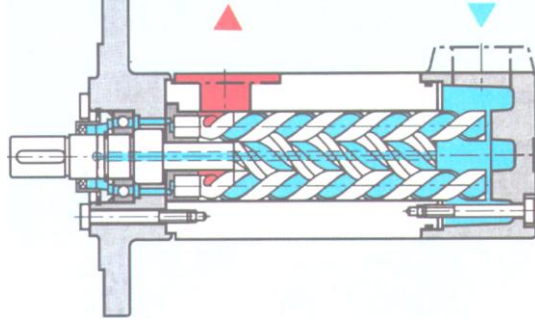
Çalışma prensipleri içten dişlilere benzer. Dişlilerin her ikisi de aynı yönde dönmektedir. İçten dişli pompadan farklı olarak gövdede bulunan hilal şeklindeki çıkıntı ortadan kaldırılmıştır. Emiş ağzından emilen sıvı basma tarafına basılmaktadır. İçteki dişlinin diş sayısı dıştakinden bir eksiktir. Pompa hacmi küçüktür. Sessiz çalışır ve arıza yapma riski zayıftır.



Şekil 3.3. Gerotor pompa (<http://extreme.infomagic.net/static/SmGerotor.jpeg>)

3.1.4. Vidalı pompalar

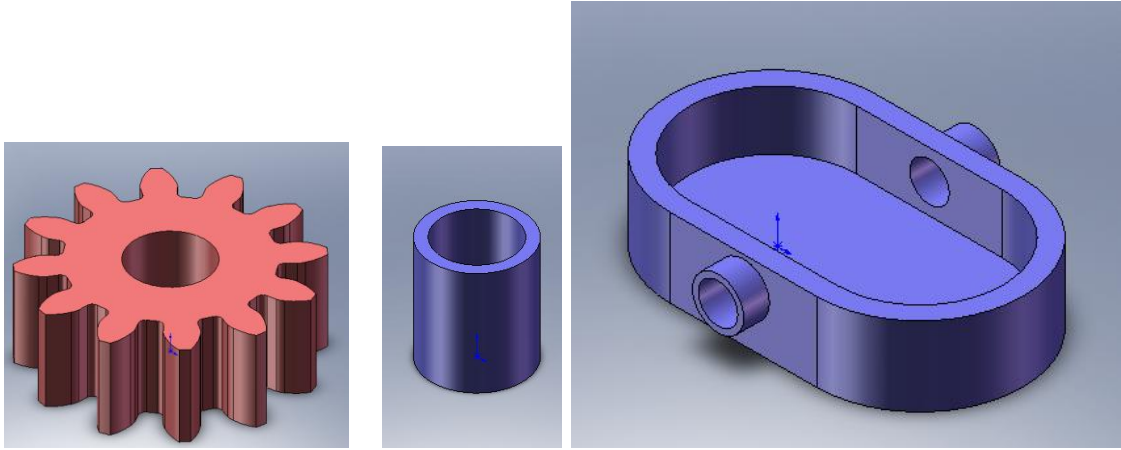
Sıvı akımı dönme eksenine doğrultusunda gerçekleşir. Burada sıvı birbirleriyle temas halindeki vidaların helisel kanalları arasında oluşan kapalı oda içinde hareket ederek emiş ağzından basma ağzına basılır. Burada rotorlar gövdeyle de temas halindedir. İçten dişli pompalar gibi vidalı pompalar da oldukça sessiz çalışırlar. Büyük debi ve yüksek basınç sağlayabilirler.



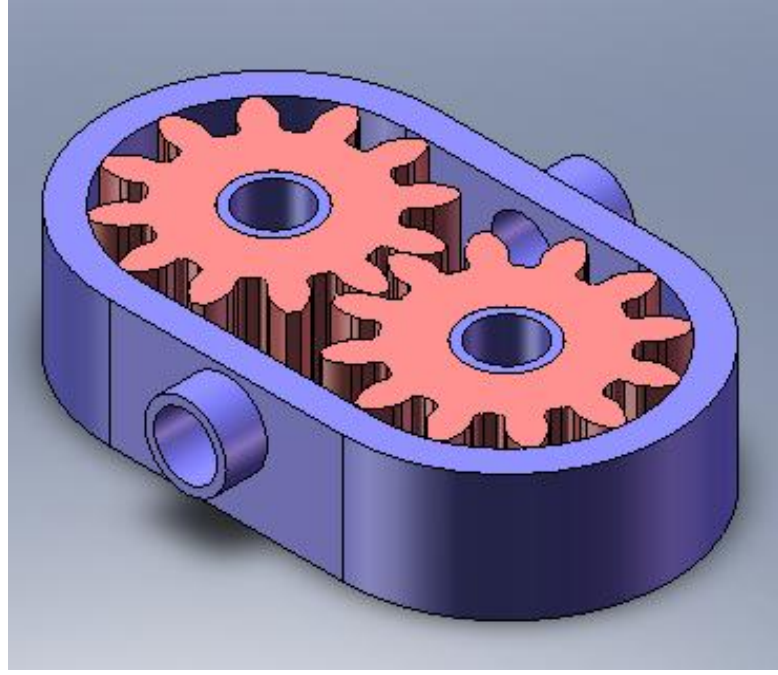
Şekil 3.4. Vidalı pompa (MEGEP, 2007)

3.2. Dişli Pompa Ana Elemanları

Bir dişli pompanın ana elemanları gövde, dişliler, pompa mili, mil yatakları ve sistemi dış etkilerden koruyan bir kapaktan oluşur.



Şekil 3.5. Dişli pompa ana elemanları



Şekil 3.6. Dişli pompanın gösterimi

3.3. Dişli Pompa Tasarımında Kullanılan Formüller

Dişli pompa tasarımında aşağıdaki donelerin bilinmesi gereklidir:

- Devir sayısı
- Basma yüksekliği
- Basılacak sıvının özgül ağırlığı
- Diş sayısı
- Modül

Diş hatvesinin hesaplanması;

$$t = m \cdot \pi \quad (3.1)$$

Diş yüksekliğinin hesaplanması;

$$h = 2.166 \cdot m \quad (3.2)$$

Taksimât dairesi çapının hesaplanması;

$$d_t = m \cdot z \quad (3.3)$$

Diş üstü dairesi çapının hesaplanması;

$$d_a = m \cdot (z + 2) \quad (3.4)$$

Temel dairesi çapının hesaplanması;

$$d_g = d_t \cdot \cos \alpha_0 \quad (3.5)$$

Taban dairesi çapının hesaplanması;

$$d_f = m \cdot (z - 2.4) \quad (3.6)$$

Diş genişliğinin hesaplanması;

$$s_0 = \pi \cdot m / 2 \quad (3.7)$$

Dişler arası açının hesaplanması;

$$\theta = 360 / z \quad (3.8)$$

Tek dişin açısının hesaplanması;

$$\alpha_t = 360 / 2 / z \quad (3.9)$$

Üst yayın hesaplanması;

$r_1 \rightarrow z$ değerine göre çizelgeden bulunur.

Alt yayın hesaplanması;

$r_2 \rightarrow z$ değerine göre çizelgeden bulunur.

Çizelge 3.1. Düzeltilmiş daire açınımlı profil diş yaylarının çiziminde kullanılan r_1 ve r_2 yarıçaplarının değerini gösteren çizelge

z	r1	r2	z	r1	r2	z	r1	r2
10	2.28M	0.69M	20	3.32M	1.89M	30	4.06M	2.76M
11	2.40M	0.83M	21	3.41M	1.98M	31	4.13M	2.85M
12	2.51M	0.96M	22	3.49M	2.06M	32	4.20M	2.93M
13	2.62M	1.09M	23	3.57M	2.15M	33	4.27M	3.01M
14	2.72M	1.22M	24	3.64M	2.24M	34	4.33M	3.09M
15	2.82M	1.34M	25	3.71M	2.33M	35	4.39M	3.16M
16	2.92M	1.46M	26	3.78M	2.42M	36	4.45M	3.23M
17	3.02M	1.58M	27	3.85M	2.50M	37-40	4.20M	
18	3.12M	1.69M	28	3.92M	2.59M	41-45	4.63M	
19	3.22M	1.79M	29	3.99M	2.67M	46-51	5.06M	

Çizelge 3.2. Modül (M) Standartları (DIN 780)

Modül (M) Standartları (DIN 780)							
0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1.25	1.5	1.75	2	2.25	2.5	2.75	3
3.25	3.5	3.75	4	4.5	5	5.5	6
6.5	7	8	9	10	11	12	13
14	15	16	18	20	22	24	27
30	33	36	39	42	45	50	55

Bir dişte taşıma miktarının hesaplanması;

$$V_a = \frac{\pi \cdot d_t \cdot m \cdot b}{z} \quad (3.10)$$

Pompanın bütün taşıma miktarının hesaplanması;

$$V_g = 2 \cdot z \cdot V_a \quad (3.11)$$

Debinin hesaplanması;

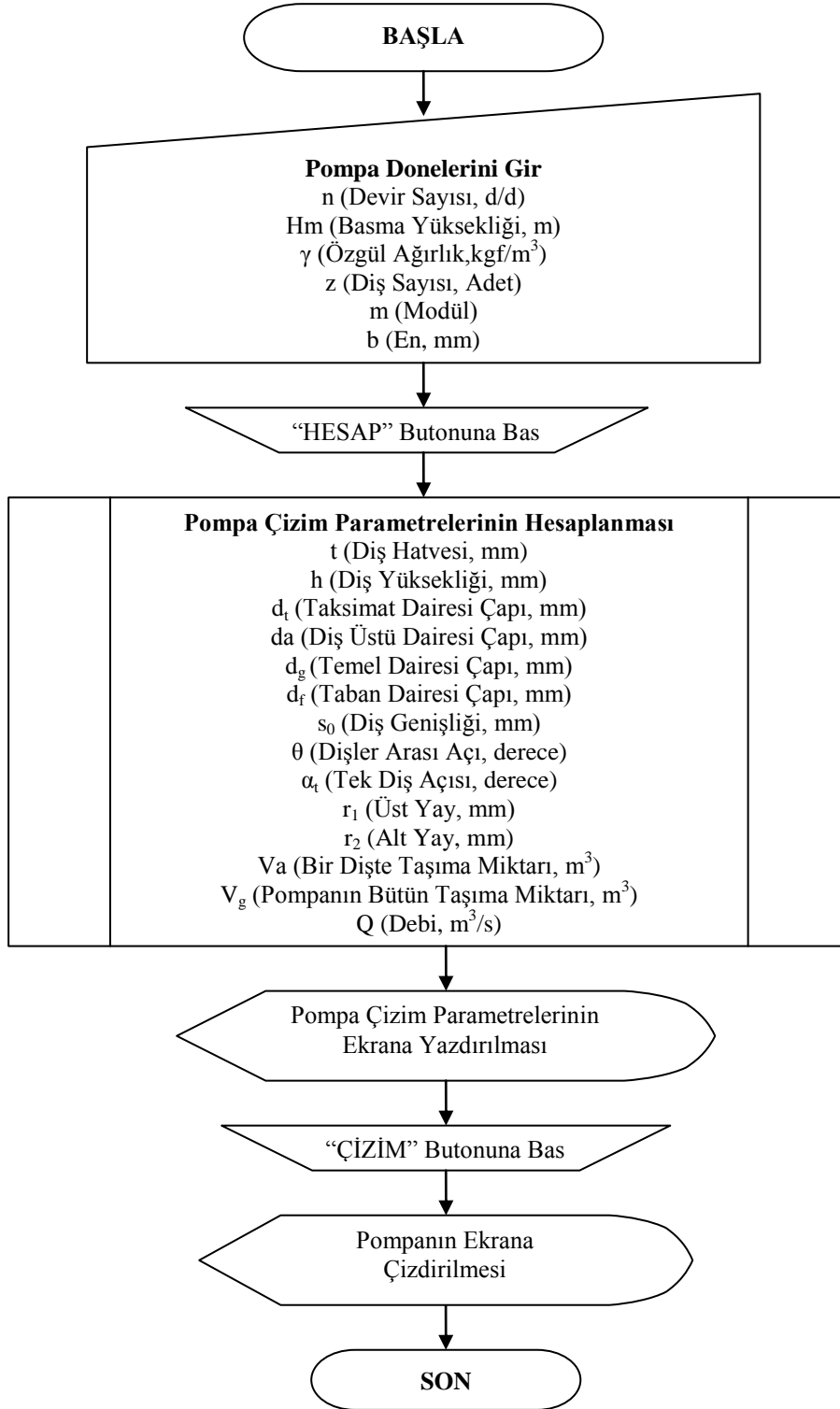
$$Q = V_g \cdot n \quad (3.12)$$

Dişlilerin çizimi;

- Eş merkezli d_a , d_b , d_g ve d_f daireleri çizilir.
- Taksimat dairesinin üzerinde bir A_1 noktası belirlenir.
- r_1 ve r_2 bulunur.
- Önceden belirlenen A_1 noktası merkez alınarak r_1 ve r_2 yayları çizilir. r_1 ve temel dairesinin çakıştığı nokta Mr_1 ; r_2 ve temel dairesinin çakıştığı nokta da Mr_2 olarak belirlenir.
- Mr_1 merkez alınarak A_1B_1 ve Mr_2 merkez alınarak A_1C_1 yayları çizilir.
- C_1 noktası ile merkez arasında taban dairesine kadar bir doğru çizilir ve alt kısma bir yarıçap verilir.
- A_1 'den s kadar uzaklıkta A_2 noktası belirlenir. $s/2$ uzaklıktan yarım diş simetriği çıkartılarak tam diş oluşturulur.
- Oluşturulan dişler belirlenen sayıya göre çoğaltılarak bir dişli çark oluşturulmuş olur.
- Dişli çark mekanizması oluşturularak diğer dişli de aynı şekilde çizilir.
- Dişlilerin bir dişli sistemi oluşturacak biçimde çizilmesinin ardından bu dişlilerin oturacağı hazne çizilir.
- Hesaplanan emme basma çaplarının çizimi de gerçekleştirilir.

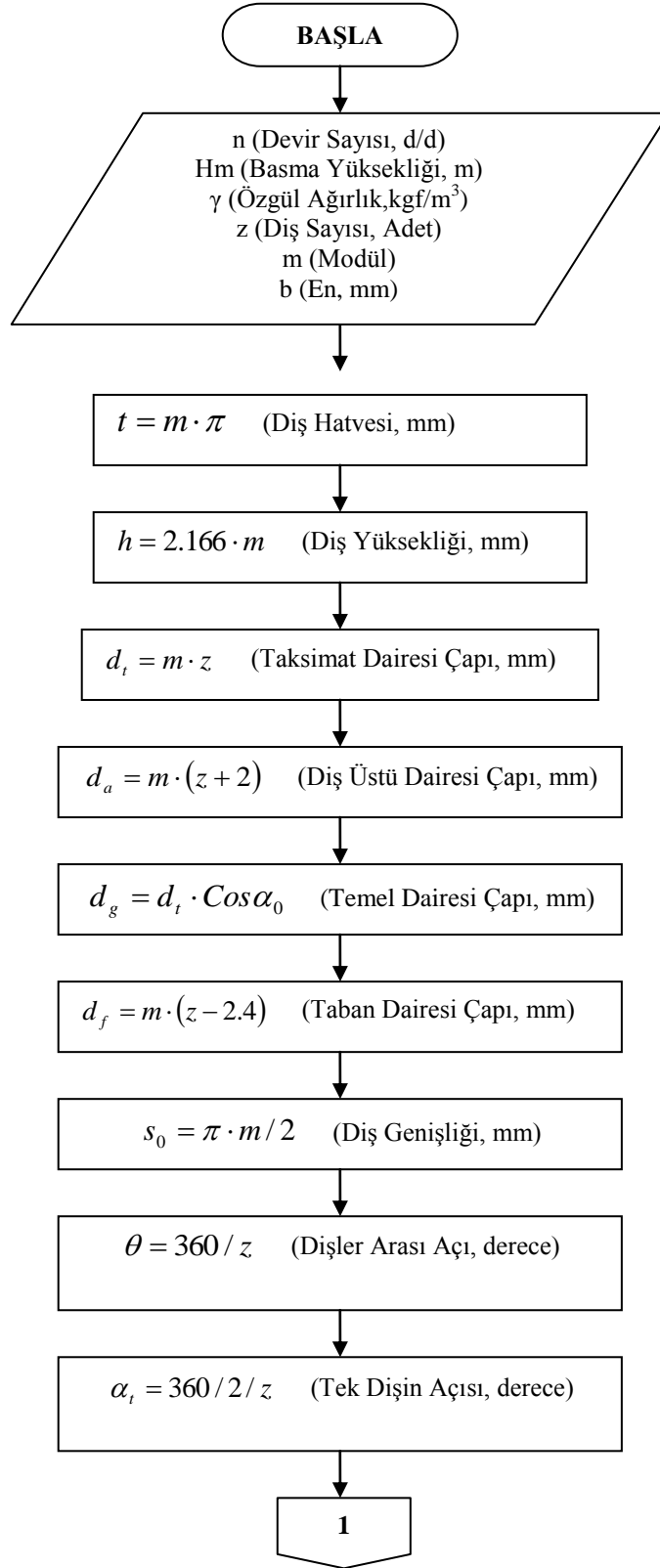
3.4. Dişli Pompa Tasarım Programı Akış Şeması

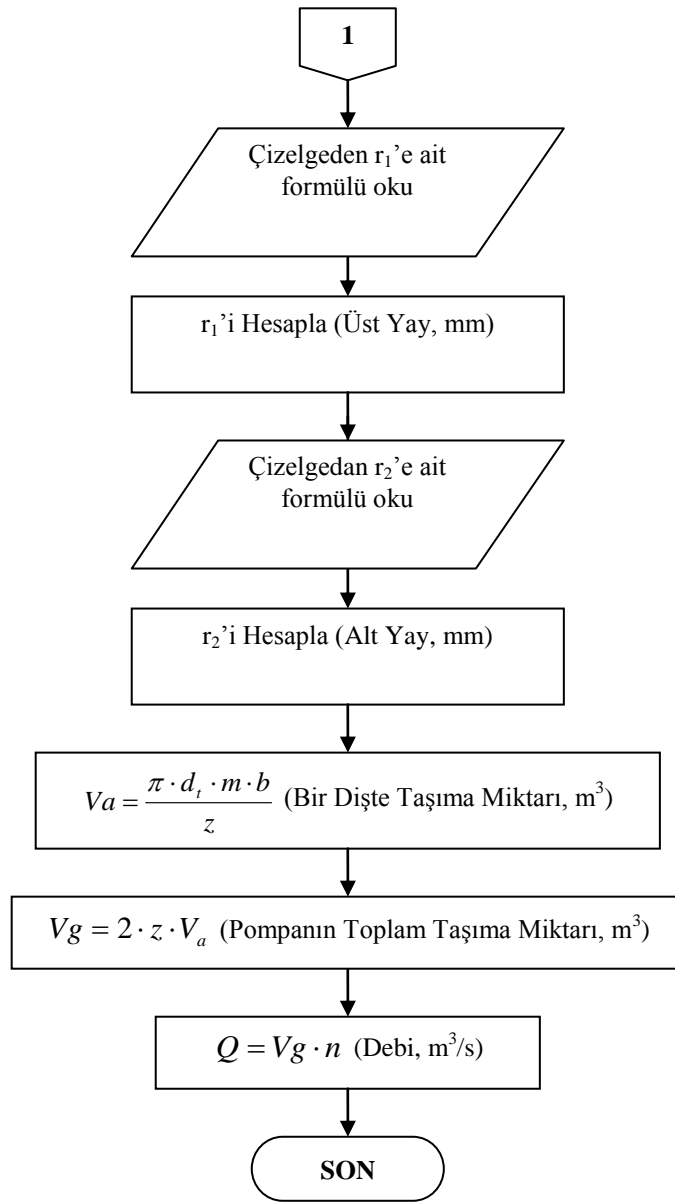
3.4.1. Ana program



Şekil 3.7. Dişli pompaya ait ana programın akış şeması

3.4.2. Alt program





Şekil 3.8. Dişli pompaya ait alt programın akış şeması

3.5. Dişli Pompa Tasarım Programının Uygulanması

Pompa doneleri:

$$n = 800 \text{ d / d (Devir Sayısı)}$$

$$Hm = 50m \text{ (Devir Sayısı)}$$

$$\gamma = 850 \text{ kgf / m}^3 \text{ (Özgül Ağırlık)}$$

$$z = 12A \text{ det (Diş Sayısı)}$$

$$m = 6 \text{ (Modül)}$$

$$b = 30mm \text{ (Yükseklik)}$$

3.5.1. Dişli pompa ile ilgili hesaplamalar

Diş hatvesinin hesaplanması;

$$t = m \cdot \pi = 6 \cdot \pi = 18.8496mm$$

Diş yüksekliğinin hesaplanması;

$$h = 2.166 \cdot m = 2.166 \cdot 6 = 12.996mm$$

Taksimat dairesi çapının hesaplanması;

$$d_t = m \cdot z = 6 \cdot 12 = 72mm$$

Diş üstü dairesi çapının hesaplanması;

$$d_a = m \cdot (z + 2) = 6 \cdot (12 + 2) = 84mm$$

Temel dairesi çapının hesaplanması;

$$d_g = d_t \cdot \cos \alpha_0 = (72\text{mm}) \cdot \cos(20) = 67.6595\text{mm}$$

Taban dairesi çapının hesaplanması;

$$d_f = m \cdot (z - 2.4) = 6 \cdot (12 - 2.4) = 57.6\text{mm}$$

Diş genişliğinin hesaplanması;

$$s_0 = \pi \cdot m / 2 = \pi \cdot 6 / 2 = 9.4248\text{mm}$$

Dişler arası açının hesaplanması;

$$\theta = 360 / z = 360 / 12 = 30^\circ$$

Tek dişin açısının hesaplanması;

$$\alpha_t = 360 / 2 / z = 360 / 2 / 12 = 15^\circ$$

Üst yayın hesaplanması;

$$r_1 = 2.51 \cdot m = 2.51 \cdot 6 = 15.06\text{mm}$$

Alt yayın hesaplanması;

$$r_2 = 0.96 \cdot m = 0.96 \cdot 6 = 5.76\text{mm}$$

Bir dişte taşıma miktarının hesaplanması;

$$V_a = \frac{\pi \cdot d_t \cdot m \cdot b}{z} = \frac{\pi \cdot (0.072m) \cdot 6 \cdot (0.030m)}{12} = 0.0033929 m^3$$

Pompanın bütün taşıma miktarının hesaplanması;

$$V_g = 2 \cdot z \cdot V_a \quad V_g = 2 \cdot 12 \cdot 0.0033929 = 0.08143 m^3$$

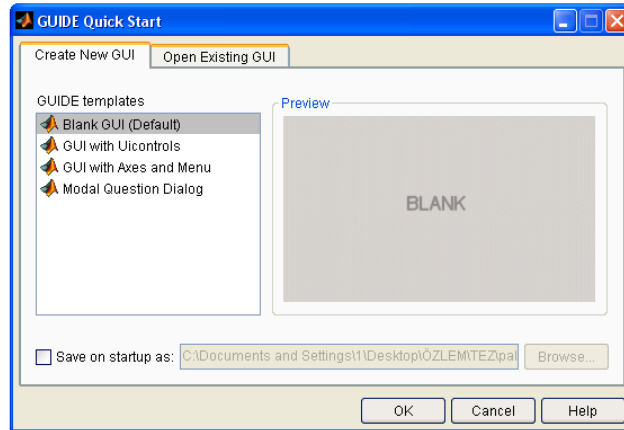
Debinin hesaplanması;

$$Q = V_g \cdot n = 0.08143 \cdot 800 = 65.144 m^3 / s$$

3.5.2. Dişli pompa tasarım yazılımının oluşturulması

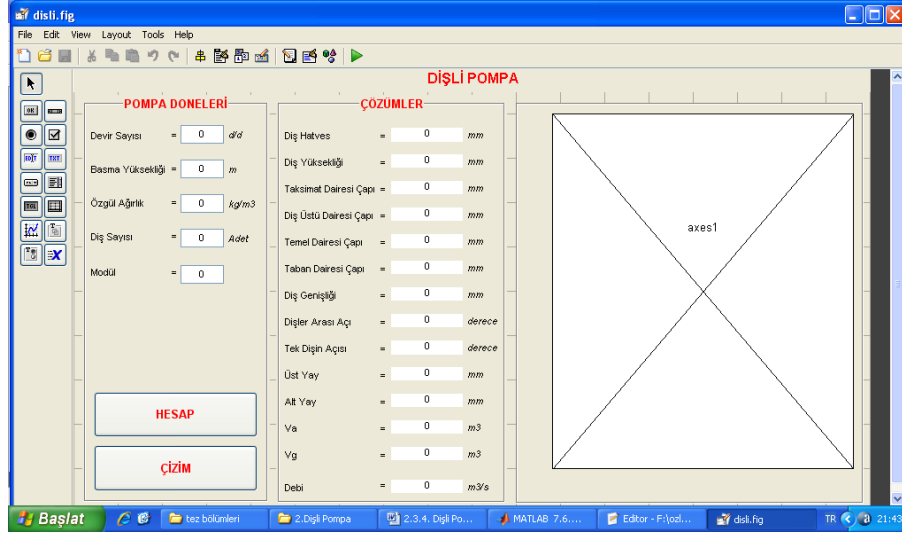
GUI'nin görüntülü kısmını oluşturmak:

- Komut Satırına “guide” yazılır.
- Gelen ekrandan Blank GUI ikonunu işaretler ve OK'ye basılır.



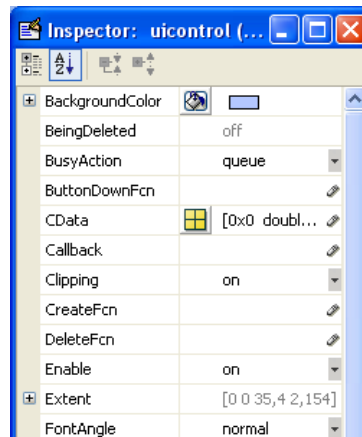
Şekil 3.9. “Guide Quick Start” ekranı

- GUI'nin nasıl görüneceğine karar veririz. Şöyle görünebilir:



Şekil 3.10. Dişli pompa GUI ön görünüm ekranı

- Sol taraftaki Menüden StaticText, EditText, Axes ve PushButton'lar sürüklenip ekrana bırakılır. Yerlerine yerleştirilir.
- StaticText'lerden biri seçilip çift tıklanır. Aşağıdaki gibi bir ekranla karşılaşılır:



Şekil 3.11. GUI uicontrol ekranı

String kısmı “POMPA DONELERİ”ne dönüştürülür. İstenirse yazı boyutu, yazı kalınlığı gibi değerler de bu kısımdan değiştirilebilir.

Aynı işlem diğer StaticTextlere de uygulanır. String kısımları Basinc, =, 0 gibi değerlerle değiştirilir.

Min	0,0	
Position	[11,8 41 36,2 2]	
SelectionHighlight	on	
SliderStep	[0,01 0,1]	
String	POMPA DONELERİ	
Style	text	
Tag	text4	
TooltipString		
UIContextMenu	<None>	

Şekil 3.12. GUI elemanlarında isimlendirme

- Çözümler bölümündeki sonuçların görüneceği StaticTextler’in String kısımlarına “0” , Tag kısımlarına da sırasıyla şu değerler yazılır:

answer_dishatvesi_staticText
 answer_disyuksekligi_staticText
 answer_taksimatdaresicapi_staticText
 answer_disustudairesicapi_staticText
 answer_temeldairesicapi_staticText
 answer_tabandairesicapi_staticText
 answer_disgenisligi_staticText
 answer_dislerarasiaci_staticText
 answer_tekdisinacisi_staticText
 answer_ustyay_staticText
 answer_altyay_staticText
 answer_va_staticText
 answer_vg_staticText
 answer_debi_staticText

- EditText'lerden birine çift tıklanıp String kısmına "0", Tag kısmına devirsayisi_editText yazılır.

- Diğer EditText'lerin String kısımlarına da "0" yazılır ancak Tag kısımlarına sırasıyla şu değerler yazılır:

basmayuksekligi_editText

ozgulagirlik_editText

dissayisi_editText

modul_editText

pompaeni_editText

kanatacisi_editText

- PushButton'lardan birine çift tıklanıp String kısmına "HESAP", Tag kısmına hesap_pushbutton yazılır.


- Diğer PushButton'un String kısımlarına da "ÇİZİM", Tag kısmına anaelemancizim_pushbutton yazılır.

- Axes'e çift tıklanıp Tag kısmına axes1 yazılır.

- Sayfa "disli" olarak kaydedilir. disli.fig ve disli.m olmak üzere iki dosya oluşur. Oluşan m.file açılır.

GUI'ye kod yazmak:

disli.m isimli m.file dosyasına Ekler'de verilen kodları yazarız.

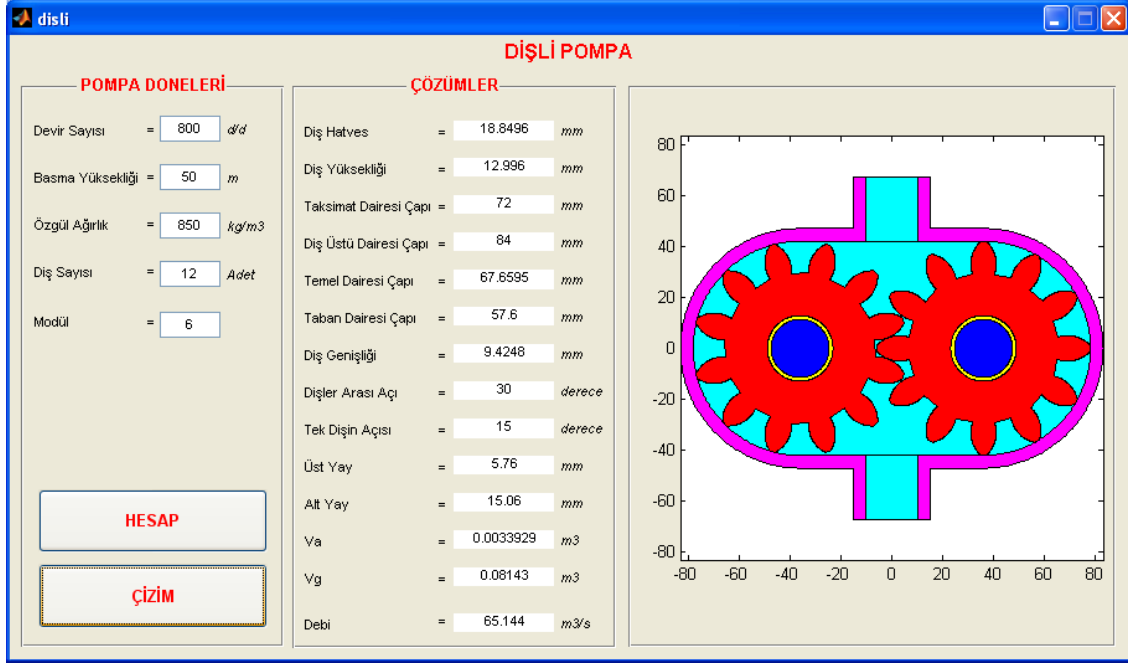
Run  ikonuna tıkladığımızda şu görünümü elde ederiz:

Şekil 3.13. Dişli pompa hesap ve çizim ekranı

Sol taraftaki pompa donelerini girerek “HESAP” butonuna bastığımızda MATLAB hesapları yapar ve sonuçlar ekranda görüntülenir.

Şekil 3.14. Dişli pompa ekranında “HESAP” butonuna basılmış durum

“ÇİZİM” butonuna bastığımızda ana elemanların çizimi axes bölümünde görüntülenir.



Şekil 3.15. Dişli pompa ekranında “ÇİZİM” butonuna basılmış durum

Program MATLAB kurulu olmayan bilgisayarda bu haliyle çalışmaz. Çalışmasını sağlamak için programı .exe uzantılı hale dönüştürmemiz gerekir. Bunun için m.file açırken komut penceresine şu komutu yazarız:

```
>>mcc – m paletli
```

Lcc compiler seçeneğini seçtiğimizde program .exe uzantılı hale gelir.

BÖLÜM 4

RADYAL SANTRİFÜJ POMPA

Sabit bir açısal hızla dönen bir çark tarafından suya verilen merkezkaç kuvvetini kullanan hidrodinamik bir pompadır. Akışkan pompa girişinden pompa çıkışına kadar radyal bir yol izlemektedir. Bir santrifüj pompada çarkın emiş tarafında oluşan vakumla sıvı çarkın kanatları arasına girer ve çarkın dönüşüyle hız kazanır. Çark kanatları ile çarkın ön ve arka profili tarafından sınırlanan kanallar arasında sıvı çarkın çıkış tarafına doğru motorla tahrik edilen milin çarkı döndürmesiyle meydana gelen santrifüj kuvvetler etkisiyle itilir. Çarkın kanatlarını hızla terk eden sıvının içerdiği kinetik enerji, salyangoz boşluğunda basınç enerjisine çevrilir.

Bütün sıvılarla kullanılabilir özelliktedir. PVC, bronz ve çeşitli alaşımli maddeler gibi malzemelerle üretimi yapılabilir.



Şekil 4.1. Santrifüj pompa görünüşü

4.1. Radyal Santrifüj Pompa Ana Elemanları

Bir radyal santrifüj pompanın ana elemanları salyangoz, çark, difüzör ve pompa milinden oluşur.

4.1.1. Salyangoz

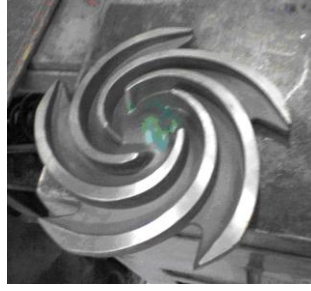
Salyangoz, çark çıkışındaki kinetik enerjiyi potansiyel enerjiye dönüştürür. Türbülansları azaltmak için pompa gövdesi salyangoz şeklinde yapılır. Salyangoz ayrıca difüzörü olmayan pompalarda çarktan çıkan suyu toplayarak difüzör görevi de görür.



Şekil 4.2. Salyangozun görünüşü

4.1.2. Çark

Mekanik enerjiyi hidrolik enerjiye dönüştüren kanatlı elemanlardır. Pompadan istenen özelliklere göre bu kanatların sayıları ve şekillerinde farklılıklar görülür. Çark sayısı kademe sayısını da gösterir. Birden fazla çarklı olan pompalar kademeli pompalardır ve çark sayısı ve kademe sayısı birbirine eşittir.



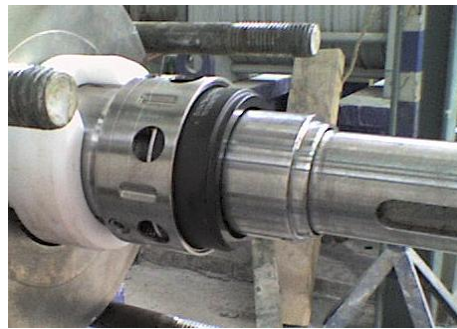
Şekil 4.3. Çarkın görünüşü

4.1.3. Difüzör (Yayıcı)

Çarktan yüksek hızla çıkan sıvının kinetik enerjisinin bir kısmını basınç enerjisine çevirmek için bazı pompalarda çark ile salyangoz arasına konulan ve üzerinde kanatlar bulunan halka şeklindeki parçadır. Difüzörlü pompaların verimi difüzörsüz olanlara göre daha fazladır.

4.1.4. Pompa Mili

Mil, pompayı tahrik eden elemandır. Motordan alınan dönüş hareketini çarka aktarır. Kademeli pompalarda taşıma görevi görür. Milin yataklaması çok hassas olmalıdır. Aksi takdirde aşınmalar çarpmalar meydana gelerek pompa kısa sürede kullanılmaz duruma düşecektir. Motordan hareketi doğrudan alabildiği gibi bir kaplin aracılığıyla da alabilir. Kaplinle bağlantı bir kama bağlantısıyla olur. Balans ayarı kama ile birlikte yapılır.



Şekil 4.4. Mil ve gövde arası bağlantının görünüşü



Şekil 4.5. Mil ve motor bağlantısı

4.2. Radyal Santrifüj Pompa Tasarımında Kullanılan Formüller

Pompa doneleri:

- Debi
- Basma Yüksekliği
- Devir Sayısı
- Basılan Sıvının Özgül Ağırlığı
- Malzeme Gerilme

Tanım sayısı;

$$n_q = \frac{n \cdot \sqrt{Q}}{H^{3/4}} \quad (4.1)$$

n_q değerine göre pompa tipi belirlenir.

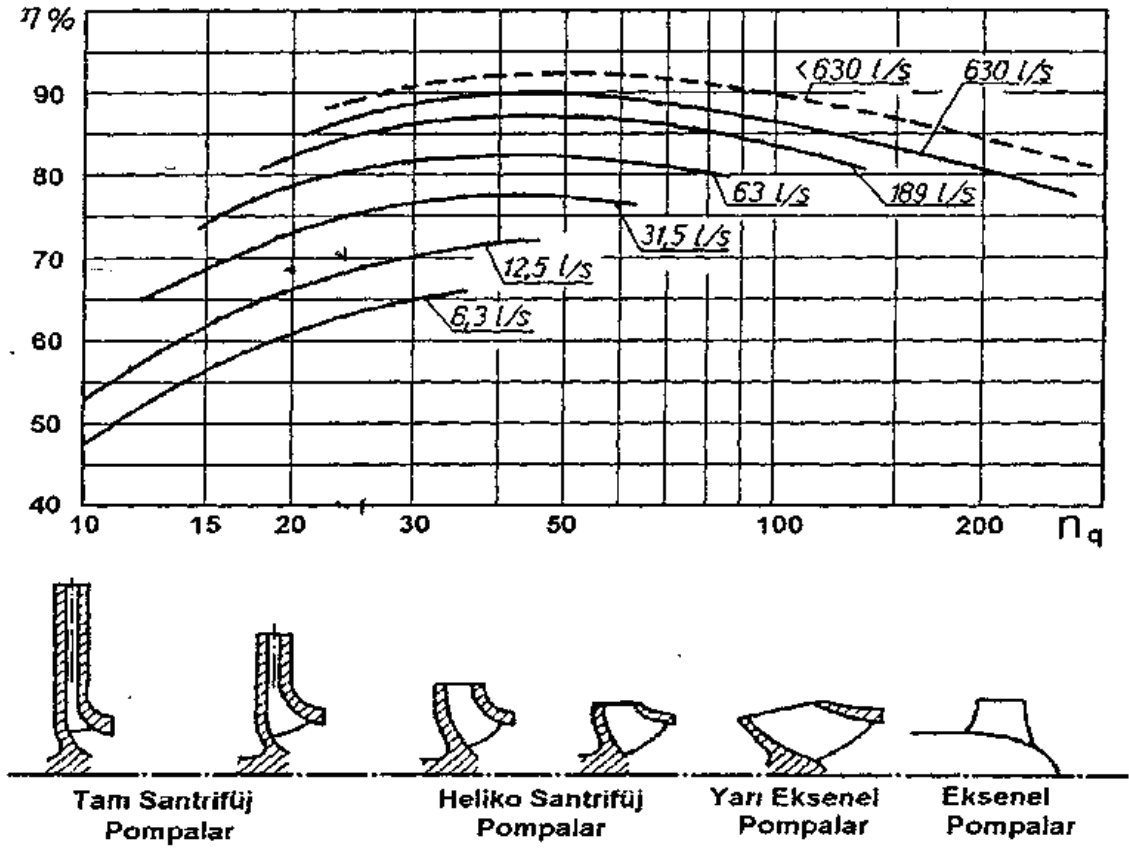
Özgül hız;

$$n_s = 3,65 \cdot n_q \quad (4.2)$$

Özgül hız pompa çark tipini belirler. n_s 'nin 45'ten küçük olması halinde pompa kademeli yapılıır.

Pompa verimi;

Pompa verimi özgül hıza bağlı olarak bulunur.



Şekil 4.6. Özgül hız-pompa verim diyagramı (Pancar, 2007)

Hidrolik verim;

$$\eta_{hidrolik} = 1 - \frac{0,071}{Q^{0,25}} \quad (4.3)$$

Mekanik verim;

Genel bir yaklaşımla pompanın mekanik verimi %96 civarında kabul edilir.

Volumetrik verim ve kaçak debi;

η_v ,volumetrik verim Q_k ise kaçak debi olmak üzere

$$\eta_{volumetrik} = \eta_{pompa} / \eta_{hidrolik} \times \eta_{mekanik} \quad (4.4)$$

$$Q' = Q / \eta_{volumetrik} \quad (4.5)$$

Pompa mil gücü hesabı;

$$P_{mil} = \frac{\gamma \cdot Q \cdot H_m}{75 \cdot \eta_{pompa}} \quad (4.6)$$

Çizelge 4.1. Emniyet Katsayısı

α emniyet sayısı	Pompa mili gücü
1,20 – 1,30	$P_{mil} < 5 \text{ BG}$
1,10 – 1,20	$5 \text{ BG} < P_{mil} < 25 \text{ BG}$
1,05 – 1,10	$P_{mil} > 25 \text{ BG}$

Motor gücü hesabı;

$$P_{motor} = \alpha \times P_{mil} \quad (4.7)$$

Mil çapı hesabı;

$$d_{mil} = \sqrt[3]{\frac{360000 \times P_{motor}}{n \times \tau}} \quad (4.8)$$

Burada, τ malzemenin kayma gerilmesidir.

Standart mil çapları 15-20-25-30-35-40-50-60-70-80-90 mm'dir. Mil çapı standart değere göre belirlenir.

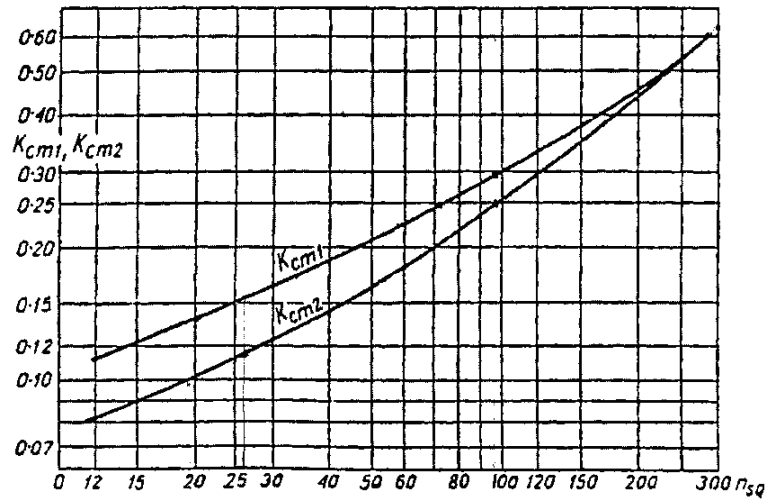
Çarkın geometrik büyüklükleri ve hız üçgenleri;

Kesite dik hız bileşenleri;

$$C_{m1} = K_{cm1} \sqrt{2Hg} \quad (4.9)$$

$$C_{m2} = K_{cm2} \sqrt{2Hg} \quad (4.10)$$

K_{cm1} ve K_{cm2} hız katsayıları özgül hıza bağlı olarak bulunur.



Şekil 4.7. K_{cm1} ve K_{cm2} diyagramı (Pancar, 2007)

Çark giriş geometrik büyüklükleri ve hız üçgeni;

Suyun kanata giriş hızı;

$$C_{m1} = K_{cm1} \sqrt{2 \cdot H \cdot g} \quad (4.11)$$

Akışkanın çarka girmeden önceki hızı;

$$C_{0m} = (0,90 \dots 0,95) \times C_{m1} \quad (4.12)$$

Giriş kesit alanı;

$$A_0 = Q' / C_{0m} \quad (4.13)$$

Göbek çapı;

$$D_g = 1,4 \times D_{mil} \quad (4.14)$$

Göbek kesit alanı;

$$A_g = \frac{\pi \times D_g^2}{4} \quad (4.15)$$

Toplam giriş kesit alanı;

$$A_{top} = A_0 + A_g \quad (4.16)$$

Çarka girişteki toplam kesit alanının dış çapı;

$$d_0 = \sqrt{\frac{4A_{top}}{\pi}} \quad (4.17)$$

Kanat üzerindeki orta akım iplikçığının çapı;

$$D_1 = \sigma_s \cdot d_0 \quad (4.18)$$

$\sigma_s = (0,9-0,95) \rightarrow$ Schultz katsayısı

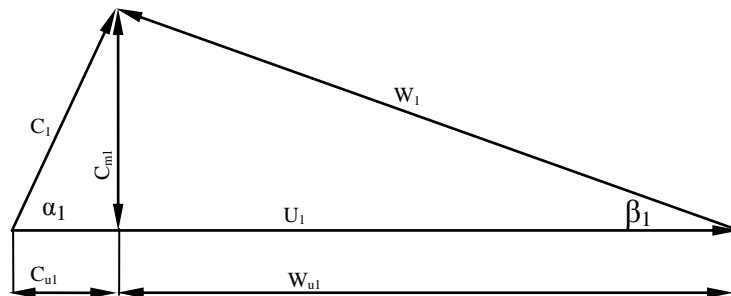
Çark girişindeki akışkan çevresel hızı;

$$U_1 = \frac{D_1 \cdot n \cdot \pi}{60} \quad (4.19)$$

Çark giriş kanat açısı;

$$\tan \beta_1 = \frac{C_{m1}}{U_1} \quad (4.20)$$

Giriş hız üçgeni;



Şekil 4.8. Giriş hız üçgeni

$$W_1 = C_{m1} / \sin \beta_1 \quad (4.21)$$

$$W_{u1} = C_{m1} / \tan \beta_1 \quad (4.22)$$

$$C_{u1} = U_1 - W_{u1} \quad (4.23)$$

$$C_1 = \sqrt{C_{m1}^2 + C_{u1}^2} \quad (4.24)$$

$$\tan \alpha_1 = C_{m1} / C_{u1} \quad (4.25)$$

Çark girişi kanat dış ve iç özellikleri;

Dış çapı;

$$D_{1d} = d_0 + 0,75 \quad (4.26)$$

İç çapı;

$$D_{1i} = 2D_1 - D_{1d} \quad (4.27)$$

Dış çevresel hız;

$$U_{1d} = (\pi \cdot D_{1d} \cdot n) / 60 \quad (4.28)$$

İç çevresel hız;

$$U_{1i} = (\pi \cdot D_{1i} \cdot n) / 60 \quad (4.29)$$

Dış açısı;

$$\beta_{1d} = \arctan (C_{m1} / U_{1d}) \quad (4.30)$$

İç açısı;

$$\beta_{1i} = \arctan (C_{m1} / U_{1i}) \quad (4.31)$$

Çark giriş genişliği;

$$\lambda_1 = 1 - \frac{Z(s_1 / \beta_1)}{\pi D_1} \quad (4.32)$$

$$b_1 = \frac{Q'}{\pi \cdot D_1 \cdot C_{m1} \cdot \lambda_1} \quad (4.33)$$

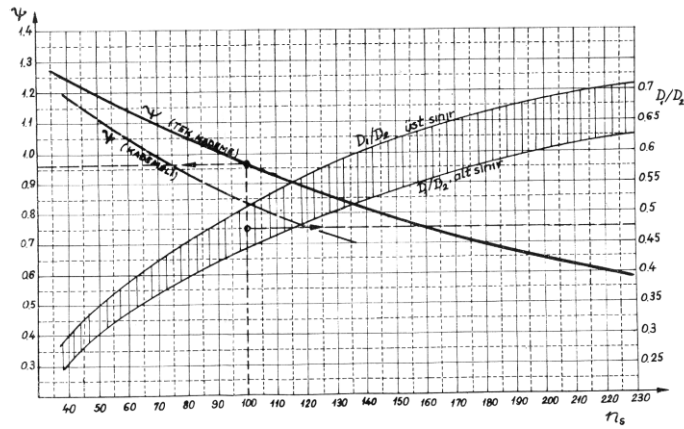
Çark çıkış geometrik büyüklükleri ve hız üçgeni;

Çark çıkışı akışkan çevresel hızı;

$$H = \psi \frac{U_2^2}{2g} \quad (4.34)$$

$$U_2 = \sqrt{\frac{2 \cdot H \cdot g}{\psi}} \quad (4.35)$$

Ψ = basınç katsayısı



Şekil 4.9. Basınç katsayısı ve D_1/D_2 oranları (Pancar, 2007)

Şekilden tek kademe için; n_s değerine karşılık Ψ okunur.

Çark çıkış çapı;

$$U_2 = \frac{\pi \cdot D_2 \cdot n}{60} \quad (4.36)$$

$$D_2 = \frac{60 \cdot U_2}{\pi \cdot n} \quad (4.37)$$

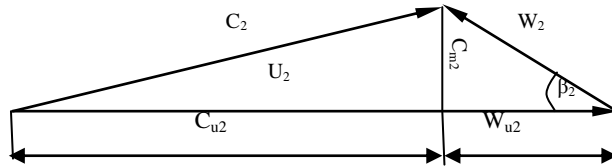
D_1/D_2 çap kontrolü;

n_s değeri için D_1/D_2 oranının maksimum ve minimum değerlerinin grafikte belirtilen aralıkta olup olmadığı kontrol edilir.

Çark çıkış kanat açısı;

Kanat açısı yaklaşık olarak 27° kabul edilir.

Çıkış hız üçgeni;



Şekil 4.10. Çıkış hız üçgeni

$$C_{m2} = K_{cm2} \sqrt{2 \cdot H \cdot g} \quad (4.38)$$

$$W_2 = C_{m2} / \sin \beta_2 \quad (4.39)$$

$$W_{u2} = C_{m2} / \tan \beta_2 \quad (4.40)$$

$$C_{u2} = U_2 - W_{u2} \quad (4.41)$$

$$C_1 = \sqrt{C_{m2}^2 + C_{u2}^2} \quad (4.42)$$

$$\tan \alpha_2 = C_{m2} / C_{u2} \quad (4.43)$$

Kanat sayısının belirlenmesi;

Kanat sayısı z

$$z = 6,5 \frac{D_2 + D_1}{D_2 - D_1} \sin \beta_m \quad (4.44)$$

$$\beta_m = \frac{\beta_1 + \beta_2}{2} \quad (4.45)$$

$$z = 6,5 \frac{D_2 + D_1}{D_2 - D_1} \sin \beta_m \quad (4.46)$$

Kanat sayısı bir üst tamsayıya tamamlanır.

Çark çıkış genişliği;

$$b_2 = \frac{Q'}{\pi \cdot D_2 \cdot \lambda_2 \cdot C_{m2}} \quad (4.47)$$

Çıkış daralma faktörü;

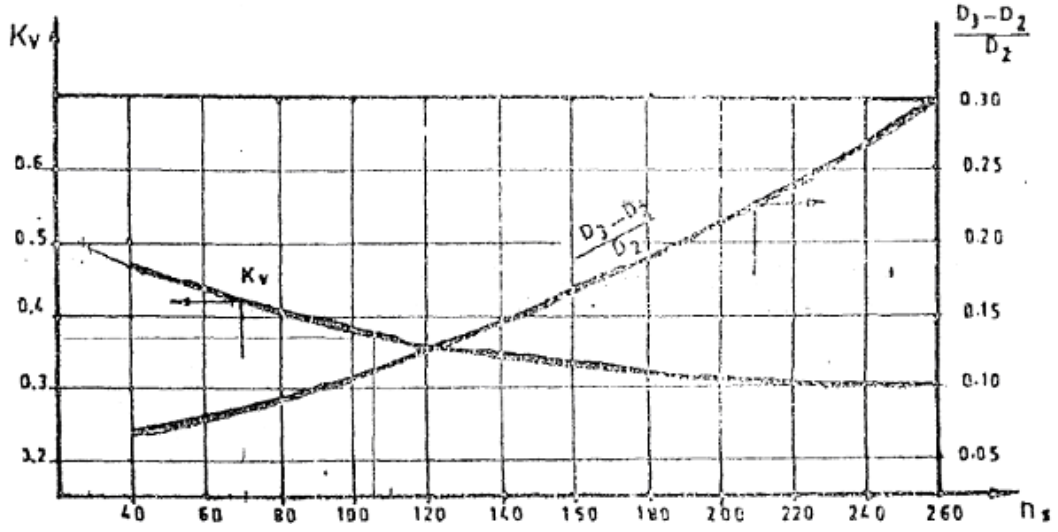
$$\lambda_2 = 1 - \frac{z \cdot (s_2 / \sin \beta_2)}{\pi \cdot D_2} \quad (4.48)$$

Salyangoz çizimi:

Eş merkezli D_2 ve D_3 daireleri çizilir. Daire merkezden itibaren eşit parçalara bölünür. Her bir bölmeden geçen debi miktarı belirtilir. Belirtilen debi miktarına göre hız formülünden her bir bölmeye ait çaplar belirtilir ve D_3 çapına teğet olarak tüm dairelerin çizimi yapılır. Daireler dıştan teğet olarak birleştirilir. $D_3/2 + D_{VIII}$ uzunluğunda çıkış borusu D_{VIII} çapından başlayarak yukarıya doğru $7^\circ - 10^\circ$ ölçülerinde genişleyerek çizilir. Böylece salyangoz çizilmiş olur.

Q = Pompanın içinden geçen debi

I. bölme	$Q/8$
II. bölme	$2Q/8$
III. bölme	$3Q/8$
IV. bölme	$4Q/8$
V. bölme	$5Q/8$
VI. bölme	$6Q/8$
VII. bölme	$7Q/8$
VIII. bölme	$8Q/8$



Şekil 4.11. $K_v - n_s$ ve $(D_3-D_2) / D_2 - n_s$ ilişkisi (Pancar, 2007)

$n_s \frac{D_3 - D_2}{D_2}$ bulunur ve D_2 yerine konularak D_3 bulunur.

n_s değerinden K_v bulunur.

$$V_s = K_v \cdot \sqrt{2 \cdot g \cdot H_m} \quad (4.49)$$

$$\frac{Q}{8} = V_s \frac{\pi \cdot D_I^2}{4} \quad D_I = \sqrt{\frac{Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.50)$$

$$\frac{2Q}{8} = V_s \frac{\pi \cdot D_{II}^2}{4} \quad D_{II} = \sqrt{\frac{2 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.51)$$

$$\frac{3Q}{8} = V_s \frac{\pi \cdot D_{III}^2}{4} \quad D_{III} = \sqrt{\frac{3 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.52)$$

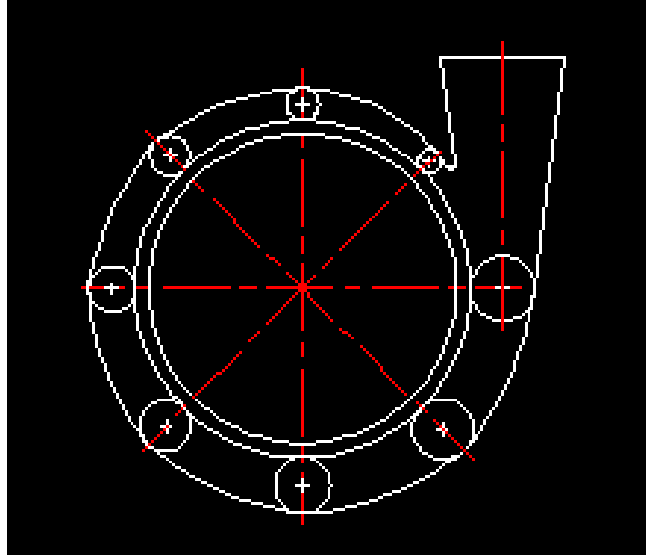
$$\frac{4Q}{8} = V_s \frac{\pi \cdot D_{IV}^2}{4} \quad D_{IV} = \sqrt{\frac{4 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.53)$$

$$\frac{5Q}{8} = V_s \frac{\pi \cdot D_V^2}{4} \quad D_V = \sqrt{\frac{5 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.54)$$

$$\frac{6Q}{8} = V_s \frac{\pi \cdot D_{VI}^2}{4} \quad D_{VI} = \sqrt{\frac{6 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.55)$$

$$\frac{7Q}{8} = V_s \frac{\pi \cdot D_{VII}^2}{4} \quad D_{VII} = \sqrt{\frac{7 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.56)$$

$$\frac{8Q}{8} = V_s \frac{\pi \cdot D_{VIII}^2}{4} \quad D_{VIII} = \sqrt{\frac{8 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \quad (4.57)$$

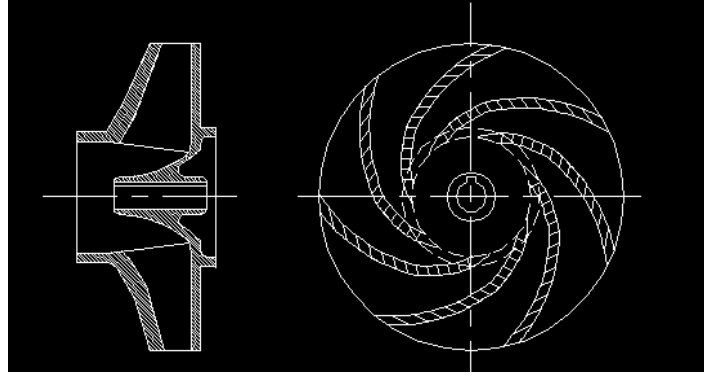
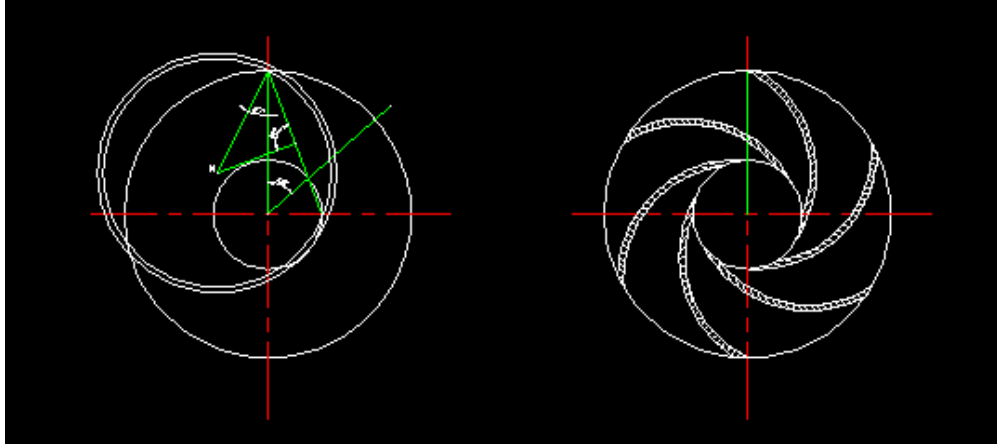


Şekil 4.12. Salyangoz

Çark kanat çizimi

Tek daire metodu;

Bu yöntemde eş merkezli D_1 ve D_2 çapları çizilir. Merkez- D_2 uzunluğunda dik bir doğru çizilir. Bu doğruya $\beta_1 + \beta_2$ kadar açı yapacak şekilde bir diğer doğru çizilir. Bu doğrunun D_1 çapını kestiği noktayla diğer doğrunun uç noktası birleştirilir. Bunun orta noktasından bir dikme çizilir. İlk çizilen doğru ile çizilen dikmenin kesişiminden M noktası bulunur. M noktasından çizilen yay ile kanat eğriliği elde edilmiş olur.



Şekil 4.13. Meridyen kesiti ve kanatlar

Nokta nokta metodu;

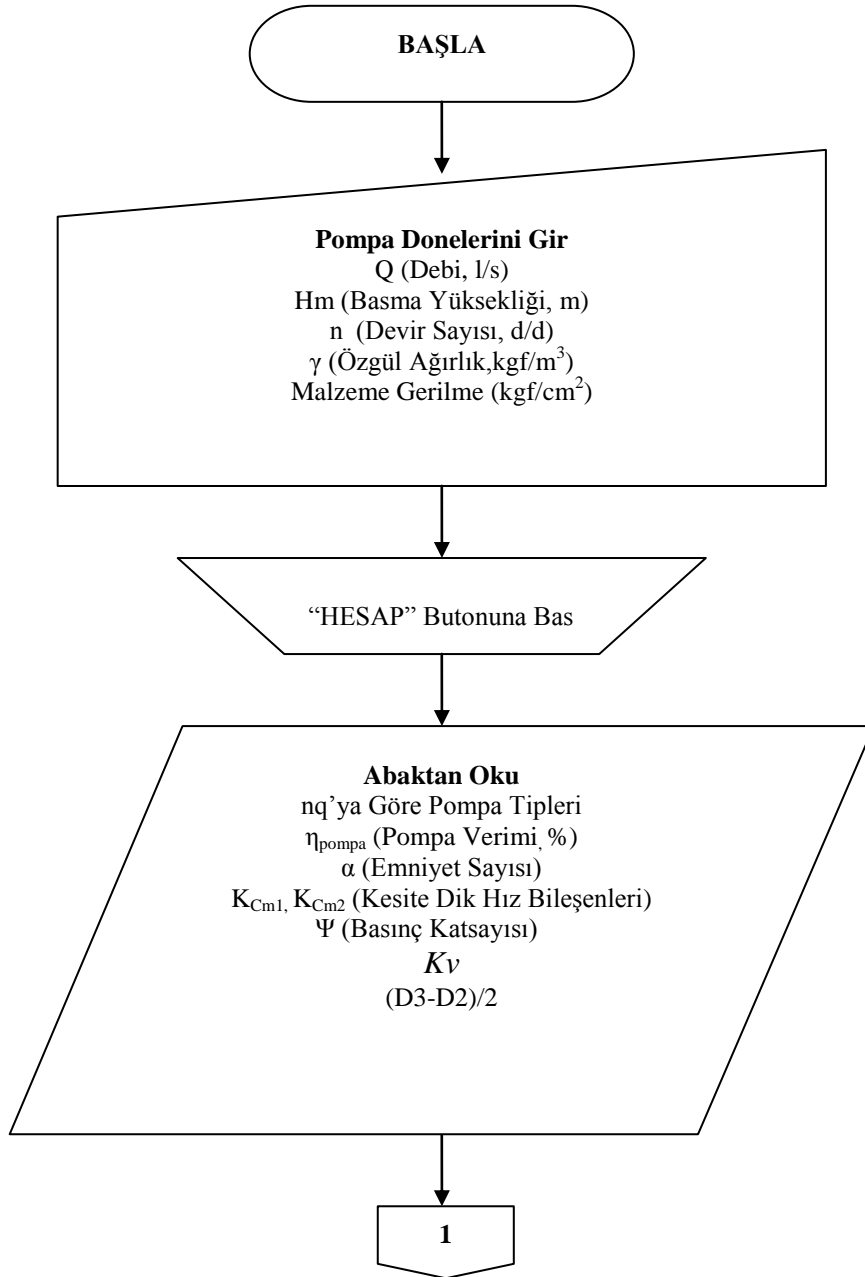
Bu metotta β_1 giriş açısı ile β_2 çıkış açısına kadar kanat boyunca nokta nokta β açılarının saptanmasıyla bu açılara uygun olarak eşit aralıklarla r yarıçapları belirlenerek kanat çizilir. Bu aralıktaki nokta sayısı dizayncıya bağlıdır. Çizimin ne kadar düzgün olmasını istiyorsak o kadar çok sayıda nokta belirlenir. Çizim için şöyle bir çizelge hazırlanır.

Çizelge 4.2. Nokta Nokta Metodu için hazırlanan çizelge

$r(\text{cm})$	$\Delta r(\text{cm})$	β	$\text{Tan } \beta$	$L=1/(r*\text{Tan } \beta)$	$\Delta a=L*\Delta R$	$\theta=180/\Delta a/\pi$	$\Sigma \theta$
----------------	-----------------------	---------	---------------------	-----------------------------	-----------------------	---------------------------	-----------------

4.3. Radyal Santrifüj Pompa Tasarım Programı Akış Şeması

4.3.1. Ana program

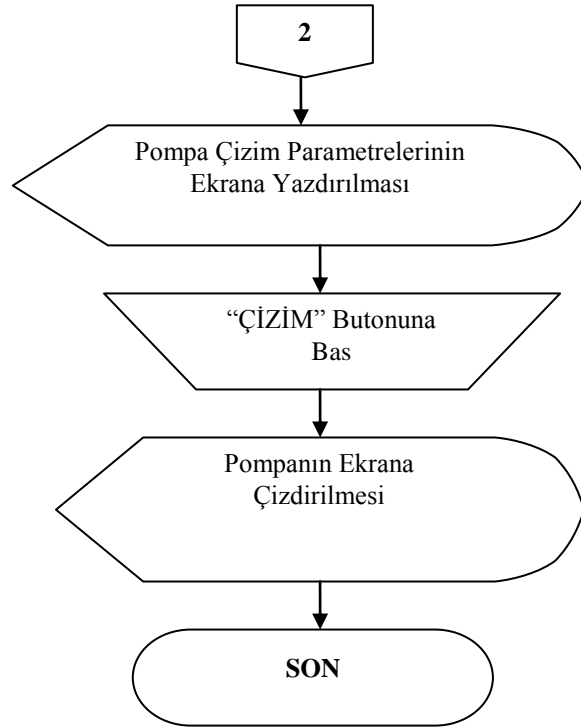


1

Pompa Çizim Parametrelerinin Hesaplanması

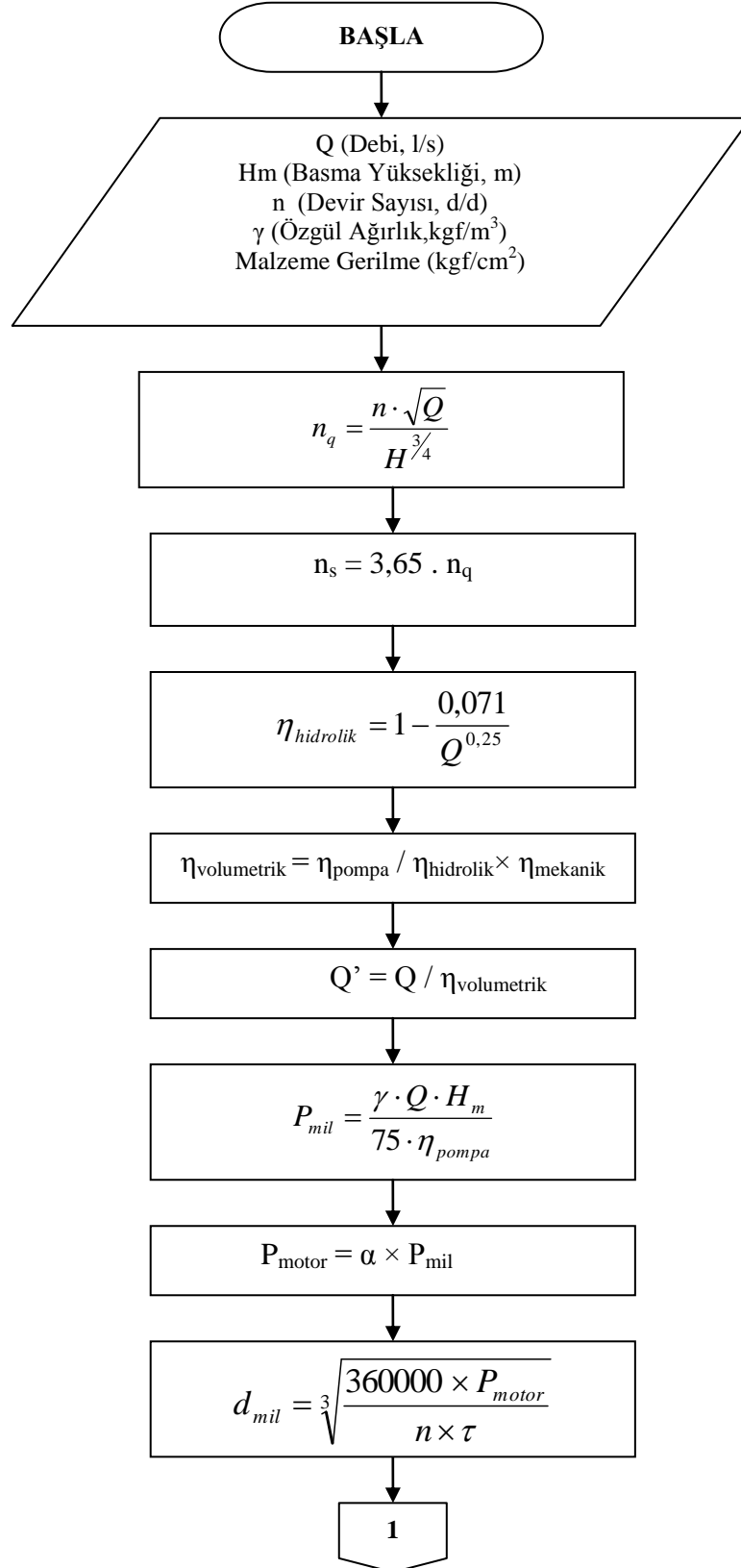
n_q (Tanım Sayısı)
 n_s (Özgül Hız, d/d)
 $\eta_{hidrolik}$ (Hidrolik Verim, %)
 $\eta_{mekanik}$ (Mekanik Verim, %)
 $\eta_{volümetrik}$ (Volümetrik Verim, %)
 Q' (Kaçak Debi, m³/s)
 P_{mil} (Pompa Mil Gücü, BG)
 P_{motor} (Motor Gücü, BG)
 d_{mil} (Mil Çapı, mm)
 C_{m1} (Suyun Kanata Giriş Hızı, m/s)
 C_{0m} (Akışkanın Çarka Girmeden Önceki Hızı, m/s)
 A_0 (Giriş Kesit Alanı, m²)
 D_g (Göbek Çapı, mm)
 A_g (Göbek Kesit Alanı, m²)
 A_{Top} (Toplam Kesit Alanı, m²)
 d_0 (Çarka Girişteki Toplam Kesit Alanın Dış Çapı, mm)
 D_1 (Kanat Üzerindeki Orta Akım İplikçiğın Çapı, mm)
 U_1 (Çark Girişindeki Akışkan Çevresel Hızı, m/s)
 β_1 (Çark Giriş Kanat Açısı, °)
 $W_1, W_{u1}, C_{u1}, C_1, \alpha_1$ (Giriş Hız Üçgen Değerleri, m/s)
 D_{1d} (Çark Giriş Kanat Dış Çap, mm)
 D_{1i} (Çark Giriş Kanat İç Çap, mm)
 U_{1d} (Çark Giriş Kanat Dış Çevresel Hız, m/s)
 U_{1i} (Çark Giriş Kanat İç Çevresel Hız, m/s)
 β_{1d} (Çark Giriş Kanat Dış Açısı, °)
 β_{1i} (Çark Giriş Kanat İç Açısı, °)
 λ_1, λ_2
 b_1 (Çark Giriş Genişliği, mm)
 H (Basma Yüksekliği, m)
 U_2 (Çark Çıkışı Akışkan Çevresel Hızı, m/s)
 D_2 (Çark Çıkış Çapı, mm)
 C_{m2} (Suyun Kanattan Çıkış Hızı, m/s)
 $W_2, W_{u2}, C_{u2}, C_2, \alpha_2$ (Giriş Hız Üçgen Değerleri, m/s)
 b_2 (Çark Çıkış Genişliği, mm)
 Z (Kanat Sayısı)
 V_s (Hız, m/s)
 $D_I, D_{II}, D_{III}, D_{IV}, D_V, D_{VI}, D_{VII}, D_{VIII}$ (Salyangoz Yardımcı Daire Çapları)

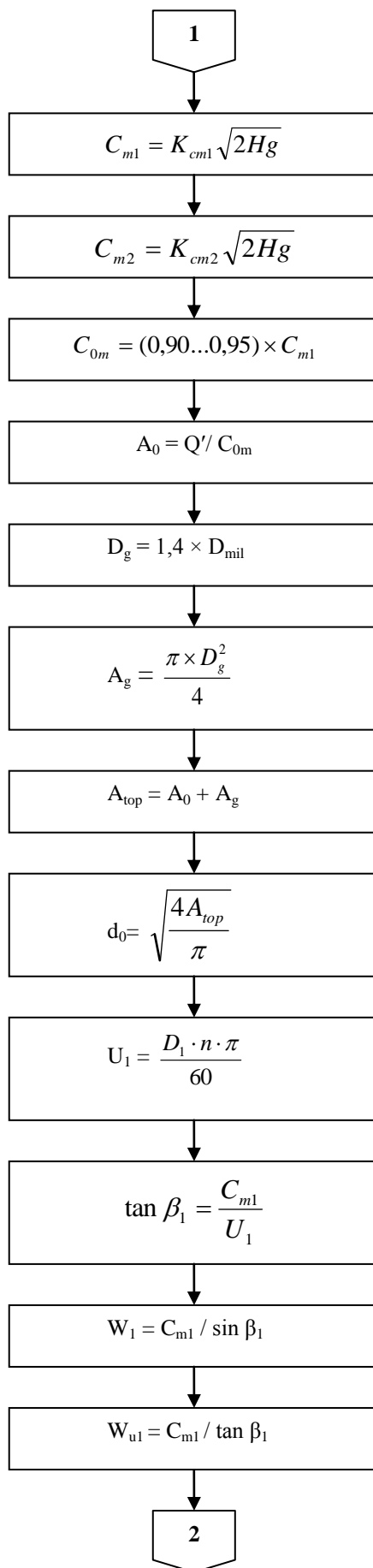
2

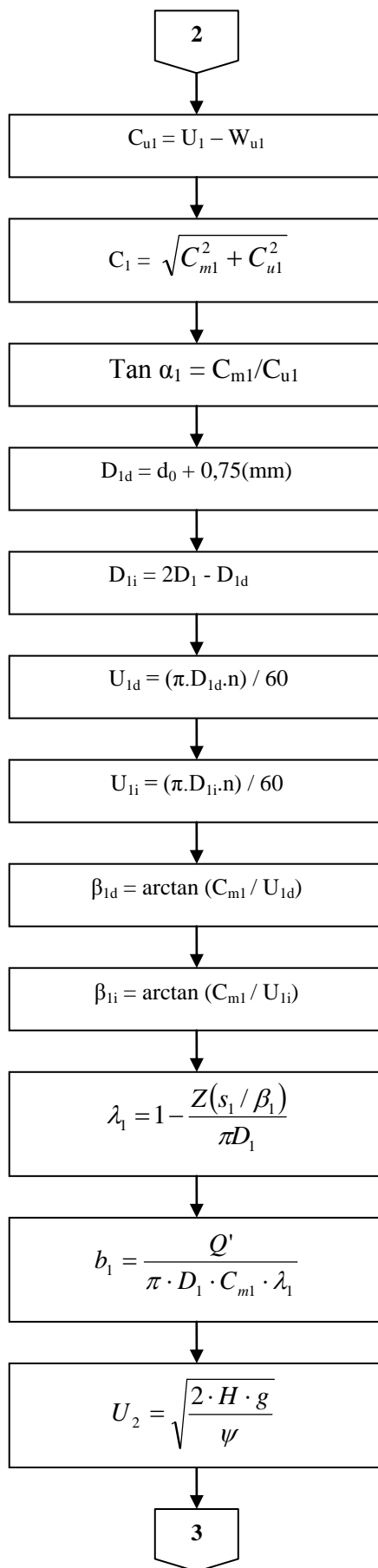


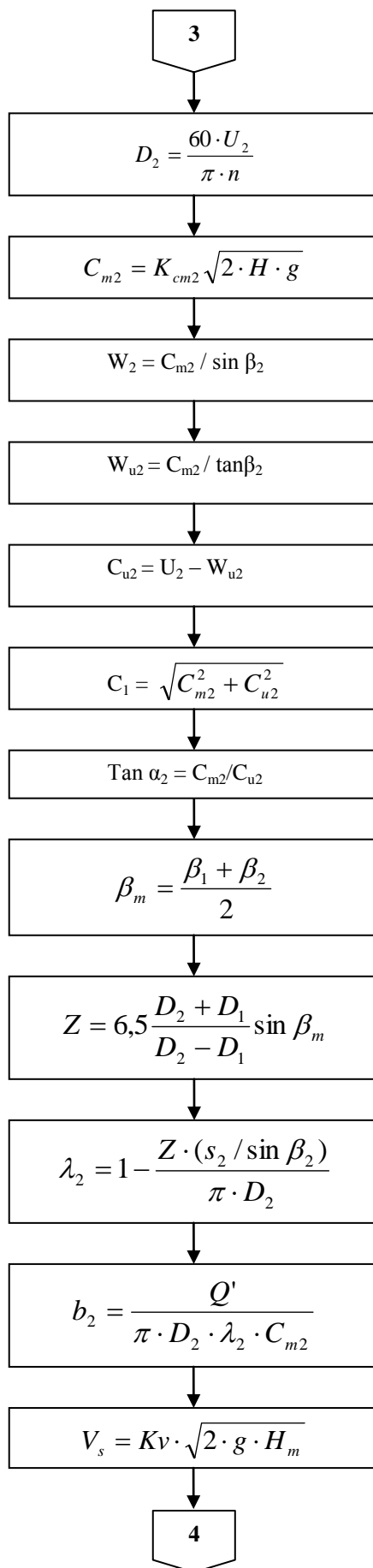
Şekil 4.14. Radyal santrifüj pompaya ait ana programın akış şeması

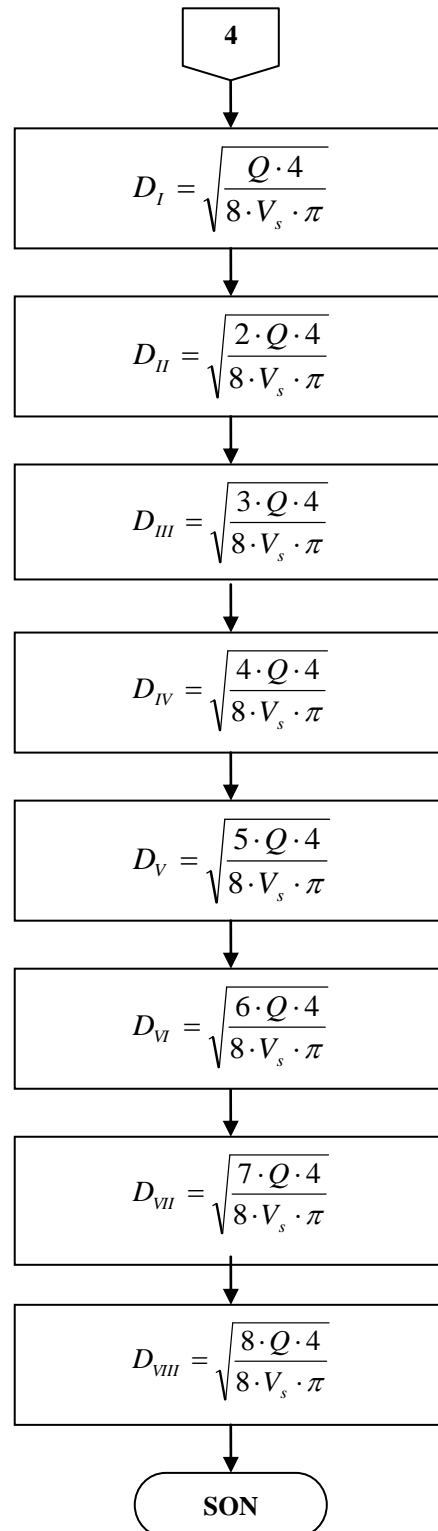
4.3.2. Alt program











Şekil 4.15. Radyal santrifüj pompaya ait alt programın akış şeması

4.4. Radyal Santrifüj Pompa Tasarım Programının Uygulanması

Pompa doneleri:

$$Q = 8 \text{ l/s (Debi)}$$

$$Hm = 30m \text{ (Basma Yüksekliği)}$$

$$n = 3000 \text{ d/d (Devir Sayısı)}$$

$$\gamma = 1000 \text{ kgf/m}^3 \text{ (Özgül Ağırlık)}$$

$$\sigma_t = 120 \text{ kgf/cm}^2 \text{ (Malzeme Gerilme)}$$

Tek kademeli

4.4.1. Radyal santrifüj pompa ile ilgili hesaplamalar

Özgül hız (n_s);

$$n_q = \frac{n\sqrt{Q}}{H^{3/4}} = \frac{3000\sqrt{0.008}}{30^{3/4}} = 20.9327$$

$$n_s = 3,65 \times 20,93 = 76,40 \text{ d/d olarak bulunur.}$$

Bulunan değere göre, pompa çarkının tipi tam santrifüj çarktır.

Pompa verimi;

$n_q = 20,93$ ve $Q = 8 \text{ l/s}$ için $\eta_{pompa} \approx \%65$ olarak okunur.

Hidrolik verim;

$$\eta_{hidrolik} = 1 - \frac{0,071}{Q^{0,25}} = 1 - \frac{0,071}{(0.008)^{0,25}} = 0.7625 \Rightarrow \eta_{hidrolik} = \%76.25$$

Mekanik verim;

Genel bir kabulde pompanın mekanik verimi %96 civarında alınır.

Volumetrik verim ve kaçak debi;

$$\eta_{volumetrik} = \eta_{pompa} / \eta_{hidrolik} \times \eta_{mekanik} = 0,65 / (0,7625 \times 0,96) = 0,887$$

$$Q' = Q / \eta_{volumetrik} = 0,008 / 0,9562 = 0,009m^3/s$$

Pompa mil gücü hesabı;

$$P_{mil} = \frac{\gamma \cdot Q \cdot H_m}{75 \cdot \eta_{pompa}} = \frac{1000 \cdot 30 \cdot 0,008}{75 \times 0,65} = 4,923BG$$

Motor gücü hesabı;

$P_{mil} = 4,5714BG$ için emniyet sayısı 1,25 alınır.

$$P_{motor} = \alpha \times P_{mil} = 1,25 \times 4,923 = 6,153BG$$

Standart değer olarak 7BG alınır.

Mil çapı hesabı;

$$d_{mil} = \sqrt[3]{\frac{360000 \times P_{motor}}{n \times \tau}} = \sqrt[3]{\frac{360000 \times 6,153}{3000 \times 120}} = 1,832cm = 18,32mm$$

Malzeme olarak, adi mil çeliği seçilsin. Adi mil çeliği için $\tau = 120 kgf/cm^2$

Standart çap 20mm'dir.

Çarkın geometrik büyüklükleri ve hız üçgenleri;

Kesite dik hız bileşenleri;

$$C_{m1} = K_{cm1} \sqrt{2Hg} \qquad C_{m2} = K_{cm2} \sqrt{2Hg}$$

K_{cm1} ve K_{cm2} hız katsayıları, özgül hıza bağlı olarak grafikten okunur.

$n_q = 20,93$ için, $K_{cm1} = 0,14$ ve $K_{cm2} = 0,11$ bulunur.

Çark giriş geometrik büyüklükleri ve hız üçgeni;

Suyun kanata giriş hızı;

$$C_{m1} = K_{cm1} \sqrt{2 \cdot H \cdot g} = 0,14 \sqrt{2 \times 30 \times 9,81} = 3.396 \text{ m/s}$$

Akışkanın çarka girmeden önceki hızı;

$$C_{0m} = (0,90 \dots 0,95) \times C_{m1} = 0,9 \times 3.396 = 3.0568 \text{ m/s}$$

Giriş kesit alanı;

$$A_0 = Q' / C_{0m} = 0,009 / 3.0568 = 0,002944 \text{ m}^2$$

Göbek çapı;

$$D_{mil} = 20 \text{ mm}$$

$$D_g = 1,4 \times D_{mil} = 1,4 \times 20 = 28 \text{ mm}$$

Göbek kesit alanı;

$$A_g = \frac{\pi \times D_g^2}{4} = \frac{\pi \times 0,028^2}{4} = 0,000615 \text{ m}^2$$

Toplam giriş kesit alanı;

$$A_{top} = A_0 + A_g = 0,0022944 + 0,000615 = 0,0035 \text{ m}^2$$

Çarka girişteki toplam kesit alanının dış çapı;

$$d_0 = \sqrt{\frac{4A_{top}}{\pi}} = \sqrt{\frac{4 \times 0,0035}{\pi}} = 0,0667 \text{ m} = 67 \text{ mm}$$

Kanat üzerindeki orta akım iplikçığının çapı;

$$D_1 = \sigma_s \cdot d_0 = 0,95 \times 67 = 64 \text{ mm}$$

$\sigma_s = (0,9-0,95) \rightarrow$ Schultz katsayısı

Çark girişindeki akışkan çevresel hızı;

$$U_1 = \frac{D_1 \cdot n \cdot \pi}{60} = \frac{0,064 \cdot 3000 \cdot \pi}{60} = 10,0531 \text{ m/s}$$

Çark giriş kanat açısı;

$$\tan \beta_1 = \frac{C_{m1}}{U_1} \Rightarrow \tan \beta_1 = \frac{3,396}{10,0531} = 0,3378 \Rightarrow \beta_1 = 20,77^\circ \rightarrow \text{İstenilen debiyi}$$

elde etmek için 22° kabul edilir.

Giriş hız üçgeni;

$$W_1 = C_{m1} / \sin \beta_1 = 3.396 / \sin 22^\circ = 9.2518 \text{ m/s}$$

$$W_{u1} = C_{m1} / \tan \beta_1 = 3.396 / \tan 22^\circ = 8.5781 \text{ m/s}$$

$$C_{u1} = U_1 - W_{u1} = 10.0531 - 8.5781 = 1.4750 \text{ m/s}$$

$$C_1 = \sqrt{C_{m1}^2 + C_{u1}^2} = \sqrt{3.396^2 + 1.4750^2} = 3.7666 \text{ m/s}$$

$$\tan \alpha_1 = C_{m1}/C_{u1} = 3.396/1.4750 = 2.439 \quad \alpha_1 = 66.94^\circ$$

Çark girişi kanat dış ve iç özellikleri;

Dış çapı;

$$D_{1d} = d_0 + 0,75$$

$$D_{1d} = 67 + 0,75 = 67.75 \text{ mm} \rightarrow 68 \text{ mm alınır}$$

İç çapı ;

$$D_{1i} = 2D_1 - D_{1d}$$

$$D_{1i} = 2 \times 64 - 68 = 60 \text{ mm}$$

Dış çevresel hız ;

$$U_{1d} = (\pi \cdot D_{1d} \cdot n) / 60$$

$$U_{1d} = (\pi \cdot 0.068 \cdot 3000) / 60 = 10.6814 \text{ m/s}$$

İç çevresel hız;

$$U_{1i} = (\pi \cdot D_{1i} \cdot n) / 60$$

$$U_{1i} = (\pi \cdot 0.060 \cdot 3000) / 60 = 9.4248 \text{ m/s}$$

Dış açısı;

$$\beta_{1d} = \arctan (C_{m1} / U_{1d})$$

$$\beta_{1d} = \arctan (3.396 / 106814.) = 17.9767^{\circ}$$

İç açısı;

$$\beta_{1i} = \arctan (C_{m1} / U_{1i})$$

$$\beta_{1i} = \arctan (3.396 / 9.4248) = 20.1901^{\circ}$$

Çark giriş genişliği;

$$\lambda_1 = 1 - \frac{Z(s_1 / \beta_1)}{\pi D_1} \quad \lambda_1 = 0.6 - 0.7$$

$$b_1 = \frac{Q'}{\pi \cdot D_1 \cdot C_{m1} \cdot \lambda_1} = \frac{0,009}{\pi \cdot 0.064 \cdot 3.396 \cdot 0.6} = 0.022m = 22mm$$

Çark çıkış geometrik büyüklükleri ve hız üçgeni;

Çark çıkışı akışkan çevresel hızı;

$$H = \psi \frac{U_2^2}{2g}$$

$$U_2 = \sqrt{\frac{2 \cdot H \cdot g}{\psi}} = \sqrt{\frac{2 \cdot 30 \cdot 9.81}{1,06}} = 23.5111m/s$$

$n_s = 76.40$ değerine karşılık $\Psi = 1,06$ grafikten okunur.

Çark çıkış çapı;

$$U_2 = \frac{\pi \cdot D_2 \cdot n}{60}$$

$$D_2 = \frac{60 \cdot U_2}{\pi \cdot n} = \frac{60 \cdot 23.5111}{\pi \cdot 3000} = 0.15m = 150mm$$

D_1/D_2 çap kontrolü;

$$\text{Min } D_1/D_2 = 0,37 \quad \text{Max } D_1/D_2 = 0,43$$

$$D_1/D_2 = 57 / 150 = 0,38$$

Bulunan çaplar oranı 0,37 ile 0,43 arasında olmasından dolayı uygundur.

Çark çıkış kanat açısı;

Kanat açısı yaklaşık olarak 27^0 kabul edilir.

Çıkış hız üçgeni;

$$C_{m2} = K_{cm2} \sqrt{2 \cdot H \cdot g} = 0,11 \sqrt{2 \cdot 30 \cdot 9.81} = 2.5420 \text{ m/s}$$

$$W_2 = C_{m2} / \sin \beta_2 = 2.5420 / \sin 27^0 = 5.5992 \text{ m/s}$$

$$W_{u2} = C_{m2} / \tan \beta_2 = 2.5420 / \tan 27^0 = 4.98 \text{ m/s}$$

$$C_{u2} = U_2 - W_{u2} = 23.5111 - 4.98 = 18.522 \text{ m/s}$$

$$C_1 = \sqrt{C_{m2}^2 + C_{u2}^2} = \sqrt{2,5420^2 + 18.522^2} = 18.52 \text{ m/s}$$

$$\tan \alpha_2 = C_{m2}/C_{u2} = 2,5420/18.52 \quad \alpha_2 = 7,8144^0$$

Kanat sayısının belirlenmesi;

Kanat sayısı Z

$$Z = 6,5 \frac{D_2 + D_1}{D_2 - D_1} \sin \beta_m$$

$$\beta_m = \frac{\beta_1 + \beta_2}{2} = \frac{22 + 27}{2} = 24,5^\circ$$

$$Z = 6,5 \cdot \frac{D_2 + D_1}{D_2 - D_1} \sin \beta_m = 6,5 \cdot \frac{150 + 64}{150 - 64} \sin 24,5 = 6.7$$

Kanat sayısı 7 seçilir.

Çark çıkış genişliği;

$$Q = V \cdot A \quad V = C_{m2} \quad A = b_2 \cdot D_2 \cdot \lambda_2 \cdot \pi \quad (\lambda_2 = \text{daralma faktörü})$$

$$\Rightarrow b_2 = \frac{Q'}{\pi \cdot D_2 \cdot \lambda_2 \cdot C_{m2}}$$

$$\text{Çıkış daralma faktörü} \quad \lambda_2 = 1 - \frac{Z \cdot (s_2 / \sin \beta_2)}{\pi \cdot D_2}$$

$s_2 = 3mm$ (kanat kalınlığı) için,

$$\lambda_2 = 1 - \frac{6 \cdot (3 / \sin 27)}{\pi \cdot 150} = 0,915$$

$$b_2 = \frac{0,009}{\pi \cdot 0,150 \cdot 0,915 \cdot 2,668} = 0,0078m = 7.82mm$$

olduğundan, $b_2 = 8mm$ seçilir.

Salyangoz çizimi:

$Q =$ Pompanın içinden geçen debi

I. bölme	$Q/8$
II. bölme	$2Q/8$
III. bölme	$3Q/8$
IV. bölme	$4Q/8$
V. bölme	$5Q/8$
VI. bölme	$6Q/8$
VII. bölme	$7Q/8$
VIII. bölme	$8Q/8$

$$n_s = 3,65 \times 20,93 = 76,40 \text{ d/d dan } K_v = 0.422$$

$$V_s = K_v \cdot \sqrt{2 \cdot g \cdot H_m} = 0.422 \cdot \sqrt{2 \cdot 9.81 \cdot 30} = 10.23 \text{ m/s}$$

$$\frac{Q}{8} = V_s \frac{\pi \cdot D_I^2}{4} \quad D_I = \sqrt{\frac{Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0111 \text{ m} = 11.1 \text{ mm}$$

$$\frac{2Q}{8} = V_s \frac{\pi \cdot D_{II}^2}{4} \quad D_{II} = \sqrt{\frac{2 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{2 \cdot 0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0157 \text{ m} = 15.7 \text{ mm}$$

$$\frac{3Q}{8} = V_s \frac{\pi \cdot D_{III}^2}{4} \quad D_{III} = \sqrt{\frac{3 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{3 \cdot 0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0193 \text{ m} = 19.3 \text{ mm}$$

$$\frac{4Q}{8} = V_s \frac{\pi \cdot D_{IV}^2}{4} \quad D_{IV} = \sqrt{\frac{4 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{4 \cdot 0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0223 \text{ m} = 22.3 \text{ mm}$$

$$\frac{5Q}{8} = V_s \frac{\pi \cdot D_V^2}{4} \quad D_V = \sqrt{\frac{5 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{5 \cdot 0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0249 \text{ m} = 24.9 \text{ mm}$$

$$\frac{6Q}{8} = V_s \frac{\pi \cdot D_{VI}^2}{4} \quad D_{VI} = \sqrt{\frac{6 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{6 \cdot 0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0273 \text{ m} = 27.3 \text{ mm}$$

$$\frac{7Q}{8} = V_s \frac{\pi \cdot D_{VII}^2}{4} \quad D_{VII} = \sqrt{\frac{7 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{7 \cdot 0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0295 \text{ m} = 29.5 \text{ mm}$$

$$\frac{8Q}{8} = V_s \frac{\pi \cdot D_{VIII}^2}{4} \quad D_{VIII} = \sqrt{\frac{8 \cdot Q \cdot 4}{8 \cdot V_s \cdot \pi}} \sqrt{\frac{8 \cdot 0.008 \cdot 4}{8 \cdot 10.23 \cdot \pi}} = 0.0315 \text{ m} = 31.5 \text{ mm}$$

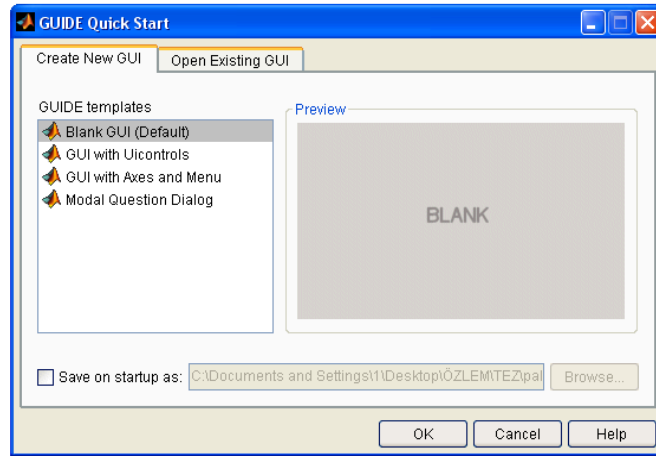
$$n_s = 3,65 \times 20,93 = 76,40 \text{ d/d} \text{ dan } \frac{D_3 - D_2}{D_2} = 0,09 \text{ ve } D_2 = 150 \text{ mm}$$

$$\frac{D_3 - D_2}{D_2} = 0,09 \quad \frac{D_3 - 150}{150} = 0,09 \quad D_3 = 163,5 \text{ mm}$$

4.4.2. Radyal santrifüj pompa tasarım yazılımının oluşturulması

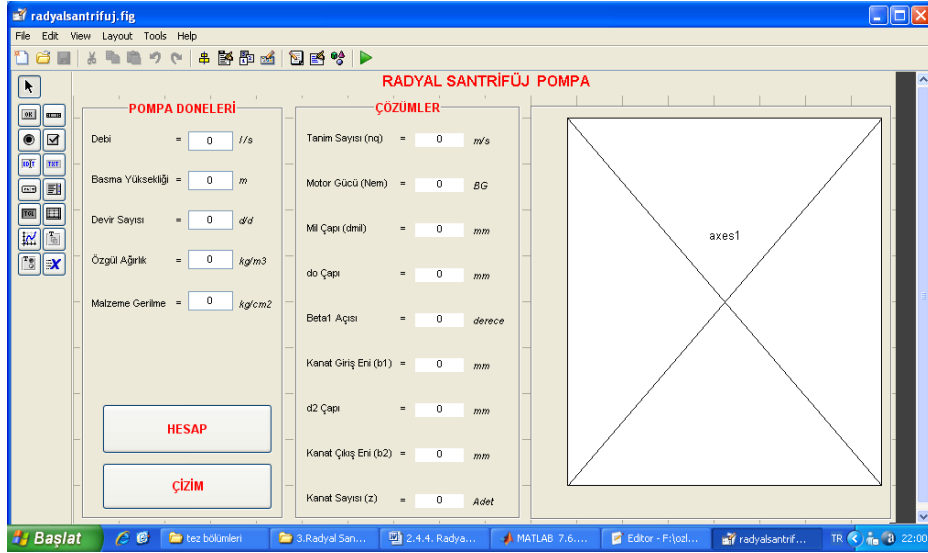
GUI'nin görüntülü kısmını oluşturmak:

- Komut Satırına “guide” yazılır.
- Gelen ekrandan Blank GUI ikonunu işaretlenir ve OK'ye basılır.



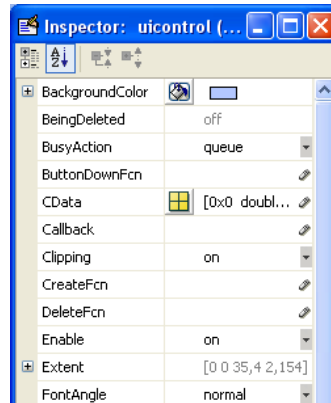
Şekil 4.16. “Guide Quick Start” ekranı

- GUI'nin nasıl görüneceğine karar verilir. Şöyle görünebilir:



Şekil 4.17. Radyal santrifüj pompa GUI ön görünüm ekranı

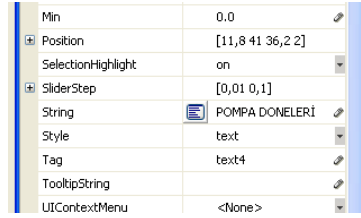
- Sol taraftaki Menüden StaticText, EditText, Axes ve PushButton'ları sürüklenip ekrana bırakılır. Yerlerine yerleştirilir.
- StaticText'lerden biri seçilip çift tıklanır. Aşağıdaki gibi bir ekranla karşılaşılır:



Şekil 4.18. GUI uicontrol ekranı

String kısmı “POMPA DONELERİ”ne dönüştürülür. İstenirse yazı boyutu, yazı kalınlığı gibi değerler de bu kısımdan değiştirilebilir.

Aynı işlemi diğer StaticTextlere de uygulanır. String kısımları şekildeki gibi Debi, =, 0, m gibi değerlerle değiştirilir.



Şekil 4.19. GUI elemanlarında isimlendirme

- Çözümler bölümündeki sonuçların görüneceği StaticTextler’in String kısımlarına “0”, Tag kısımlarına da sırasıyla şu değerler yazılır:

```
answer_tanimsayisistaticText
answer_motorgucustaticText
answer_milcapistaticText
answer_docapistaticText
answer_beta1acisistaticText
answer_kanatgirisenistaticText
answer_d2capistaticText
answer_kanatcikisenistaticText
```

- EditText’lerden birine çift tıklanıp String kısmına “0”, Tag kısmına debi_editText yazılır.

- Diğer EditText'lerin String kısımlarına da "0" yazılır ancak Tag kısımlarına sırasıyla şu değerler yazılır:

basmayuksekligi_editText
devirsayisi_editText
ozgulagirlik_editText
malzemegerilme_editText

- PushButton'lardan birine çift tıklanıp String kısmına "HESAP", Tag kısmına hesap_pushbutton yazılır.

- Diğer PushButton'un String kısmına da "CIZIM", Tag kısmına cizim_pushbutton yazılır.

- Axes'e çift tıklanıp Tag kısmına axes1 yazılır.

- Sayfa "radyalsantrifuj" olarak kaydedilir. radyalsantrifuj.fig ve radyalsantrifuj .m olmak üzere iki dosya oluşur. Oluşan m.file açılır.

GUI'ye kod yazmak:

radyalsantrifuj.m isimli m.file dosyasına Ekler'de verilen kodları yazarız.

Run  ikonuna tıkladığımızda Şekil 4.20'deki görünümü elde ederiz:

The screenshot shows the 'RADYAL SANTRİFÜJ POMPA' software interface. It is divided into three main sections: 'POMPA DONELERİ' (Pump Parameters), 'ÇÖZÜMLER' (Solutions), and a graph area.

POMPA DONELERİ (Pump Parameters):

- Debi = 0 l/s
- Basma Yüksekliği = 0 m
- Devir Sayısı = 0 d/d
- Özgül Ağırlık = 0 kg/m³
- Malzeme Gerilme = 0 kg/cm²

ÇÖZÜMLER (Solutions):

- Tanım Sayısı (nq) = 0 m/s
- Motor Gücü (Nem) = 0 BG
- Mil Çapı (dmil) = 0 mm
- do Çapı = 0 mm
- Beta1 Açısı = 0 derece
- Kanat Giriş Eni (b1) = 0 mm
- d2 Çapı = 0 mm
- Kanat Çıkış Eni (b2) = 0 mm
- Kanat Sayısı (z) = 0 Adet

Below the input fields are two buttons: 'HESAP' (Calculate) and 'ÇİZİM' (Draw). To the right is a blank graph area with axes ranging from 0 to 1.

Şekil 4.20. Radyal santrifüj pompa hesap ve çizim ekranı

Sol taraftaki pompa donelerini girerek “HESAP” butonuna bastığımızda MATLAB hesapları yapar ve sonuçlar ekranda görüntülenir.

The screenshot shows the 'RADYAL SANTRİFÜJ POMPA' software interface after the 'HESAP' button has been clicked. The 'HESAP' button is highlighted with a yellow border, and the 'ÇİZİM' button is also highlighted. The input fields now contain numerical values, and the 'ÇÖZÜMLER' section shows the calculated results.

POMPA DONELERİ (Pump Parameters):

- Debi = 8 l/s
- Basma Yüksekliği = 30 m
- Devir Sayısı = 3000 d/d
- Özgül Ağırlık = 1000 kg/m³
- Malzeme Gerilme = 120 kg/cm²

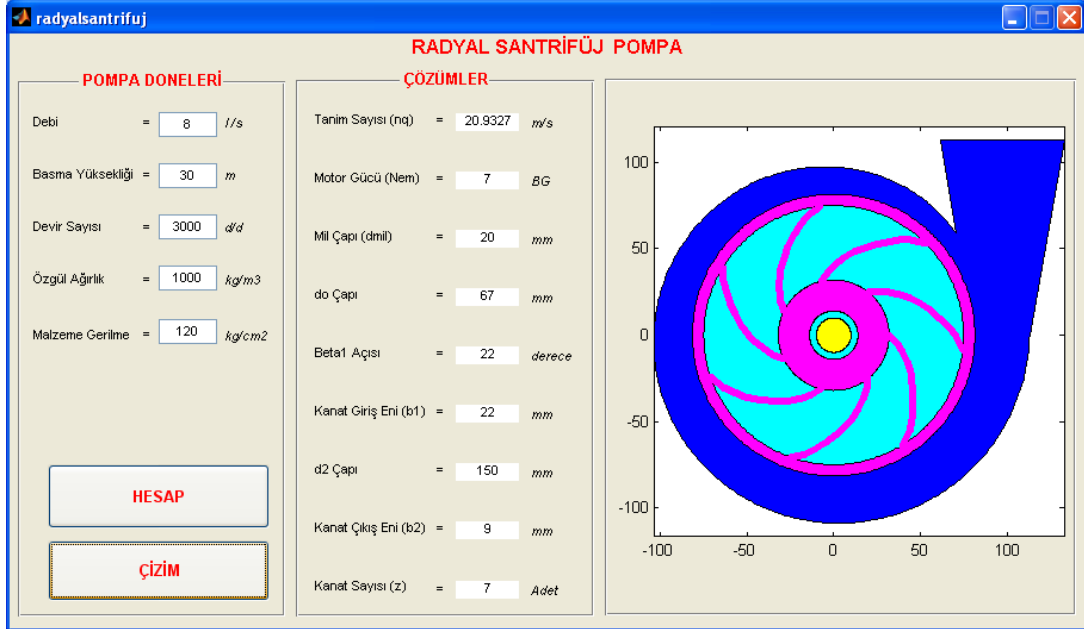
ÇÖZÜMLER (Solutions):

- Tanım Sayısı (nq) = 20.9327 m/s
- Motor Gücü (Nem) = 7 BG
- Mil Çapı (dmil) = 20 mm
- do Çapı = 67 mm
- Beta1 Açısı = 22 derece
- Kanat Giriş Eni (b1) = 22 mm
- d2 Çapı = 150 mm
- Kanat Çıkış Eni (b2) = 9 mm
- Kanat Sayısı (z) = 7 Adet

The graph area remains blank.

Şekil 4.21. Radyal santrifüj pompa ekranında “HESAP” butonuna basılmış durum

“ÇİZİM” butonuna bastığımızda ana elemanların çizimi axes bölümünde görüntülenir.



Şekil 4.22. Radyal santrifüj pompa ekranında “ÇİZİM” butonuna basılmış durum

Program MATLAB kurulu olmayan bilgisayarda bu haliyle çalışmaz. Çalışmasını sağlamak için programı .exe uzantılı hale dönüştürmemiz gerekir. Bunun için m.file açırken komut penceresine şu komutu yazarız:

```
>>mcc – m radyalsantrifuj
```

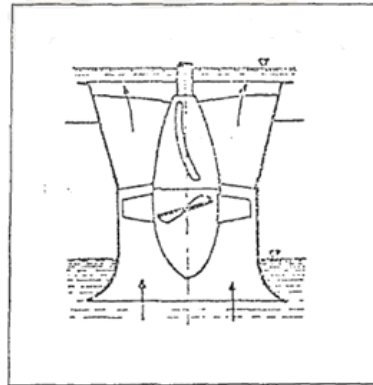
Lcc compiler seçeneğini seçtiğimizde program .exe uzantılı hale gelir.

BÖLÜM 5

EKSENEL SANTRİFÜJ POMPA

Özgül hız büyüyünce santrifüj tulumbanın çark formunun değiştiği ve akışın radyal halden aksenel hale kaydığını söylemiştik. Gerçekten çok büyük debiler için en uygun çözüm pompanın akış kesidini büyütmeektir. Ayrıca manometrik yüksekliğin büyük olması da istenmediğine göre helisel çark (Eksenel akışlı çark) en uygun çözüm olmaktadır. Bu pompalarda suyun çarkı geçmesi esnasında sürtme kayıplarını azaltmak daha büyük bir önem kazandığı için uskur türbinlerinde olduğı gibi çark açık tipte yapılır. Yani burada çark kanatlarını dıştan kavrayan bir çember mevcut değıldir (Özgür, 1972).

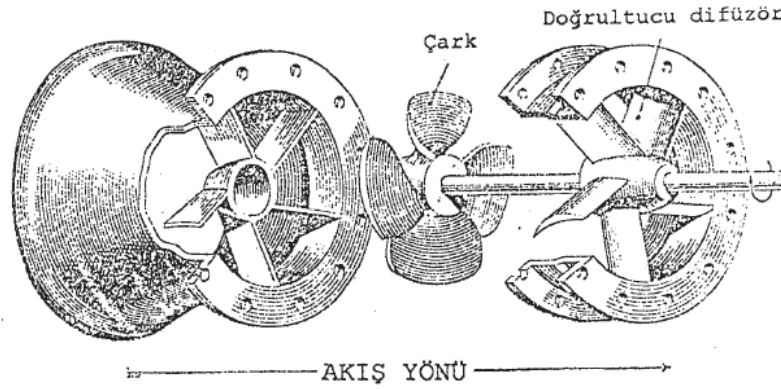
Genelde uskur pervaneli pompalar düşük basma yükseklerinde büyük hacimli su pompalamak için en iyi tercih kabul edilir. Bu pompa sersinin tipik uygulama alanları, fırtına suyu istasyonları, atıksu arıtma tesisleri, saha drenajı, sulama ve su atraksiyonlarıdır. Bu serideki pompaların kapasite aralığı 100'den 5,000 litre/saniye'ye kadardır. (<http://www.dengeteknoloji.com/atik-su-pompalari.asp>)



Şekil 4.1. Eksenel santrifüj pompa (Özgür, 1972)

5.1. Eksenel Santrifüj Pompa Ana Elemanları

Bir eksenel santrifüj pompanın ana elemanları çark, doğrultucu difüzör ve pompa milinden oluşur.



Şekil 5.2. Eksenel santrifüj pompa elemanları (Özgür, 1972)

5.2. Eksenel Santrifüj Pompa Tasarımında Kullanılan Formüller

Pompa doneleri:

- Debi
- Basma Yüksekliği
- Basılan Sıvının Özgül Ağırlığı
- Devir Sayısı

Tanım sayısı;

$$n_q = \frac{n \cdot \sqrt{Q}}{H^{3/4}} \quad (5.1)$$

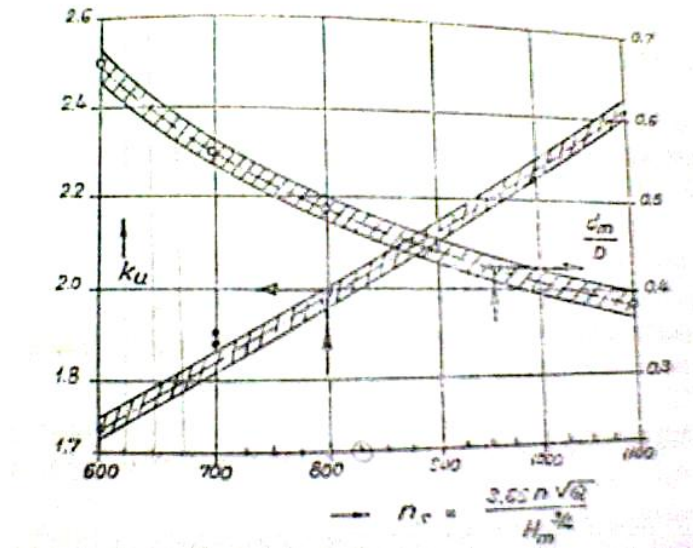
n_q değerine göre pompa tipi belirlenir.

Özgül hız;

$$n_s = 3,65 \cdot n_q \quad (5.2)$$

Şekil 4.6'ya göre pompa tipi belirlenir.

D ve d_m çap tayini;



Şekil 5.3. n_s - k_u - d_m/D diyagramı (Özgür, 1972)

$$c_m = K_U \sqrt{2 \cdot g \cdot H} \quad (5.3)$$

$$U_{du} = \frac{\pi \cdot d \cdot n}{60} \quad (5.4)$$

ifadelerinden D çapı hesaplanır. Aynı diyagramdan d_m/D oranı, seçilerek burada, d_m göbek çapı tayin edilir.

Kaçak debi;

$$Q' = 1.06 \cdot Q \quad (5.5)$$

Kesit alanı;

$$A_m = Q' / c_m \quad (5.6)$$

dh ve d2 çapı;

$$d_h = 0.45 \cdot d_2 \quad (5.7)$$

$$A_m = \frac{\pi}{4} \cdot [d_2 - (0.45 \cdot d_2)^2] \quad (5.8)$$

Kanat sayısının tespiti;

Eksenel santrifüj pompalar için kanat sayısı 3 ile 5 arasında seçilir. Büyük özgül hızlar ve küçük manometrik yükseklikler için küçük kanat sayısı alınır.

Pompa mil gücü hesabı;

$$P_{mil} = \frac{\gamma \cdot Q \cdot H_m}{75 \cdot \eta_{pompa}} \quad (5.9)$$

Motor gücü hesabı;

Çizelge 4.1'e göre emniyet katsayısı bulunur.

$$P_{motor} = \alpha \times P_{mil} \quad (5.10)$$

Kanat çiziminde kullanılan formüller;

Bulunan değerlere göre kanat çizimi için gerekli ön bilgiler elde edilmiş olur. Daha sonra kanat en az beş ayrı noktasından kesitlerine ayrılarak her bir kesit için hesaplamalar yapılır.

$$u = \frac{\pi \cdot d \cdot n}{60} \quad (5.11)$$

$$c_{u3} = \frac{g \cdot H_{th}}{u} \quad (5.12)$$

$$w_{\infty}^2 = c_m^2 + \left(u - \frac{c_{u3}}{2} \right)^2 \quad (5.13)$$

$$\tan \beta_{\infty} = \frac{c_m}{u - \frac{c_{u3}}{2}} \quad (5.14)$$

$$C_L \cdot \frac{l}{t} = \frac{2 \cdot g \cdot H_{th} \cdot c_m \cdot \cos \lambda}{w_{\infty}^2 \cdot u \cdot \sin(\beta_{\infty} + \lambda)} \quad (5.15)$$

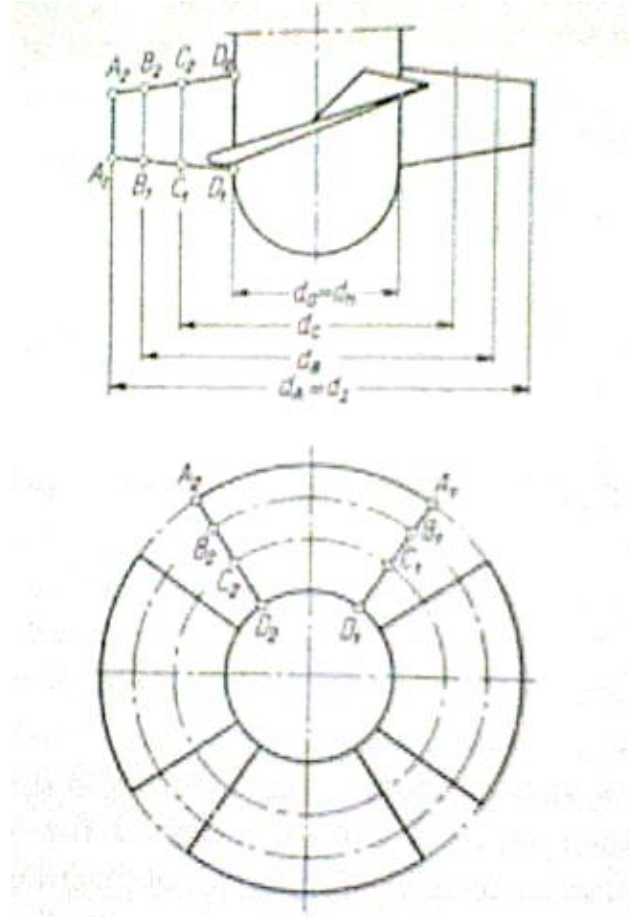
$$t = \frac{\pi \cdot d}{3} \quad (5.16)$$

$$\tan \lambda = \frac{C_D}{C_L} = 0.012 + 0.06 \cdot \frac{y_{\max}}{l} \quad (5.17)$$

$$C_L = 4.4 \cdot \frac{y_{\max}}{l} + 0.092 \cdot \alpha^0 \quad (5.18)$$

$$\Delta h = \frac{0.7 \cdot C_L \cdot W_\infty^2}{g} \quad (5.19)$$

Bulunan deęerlere ve hesaplanan açılara göre eşmerkezli olarak profiller gövde üzerine çizilir.



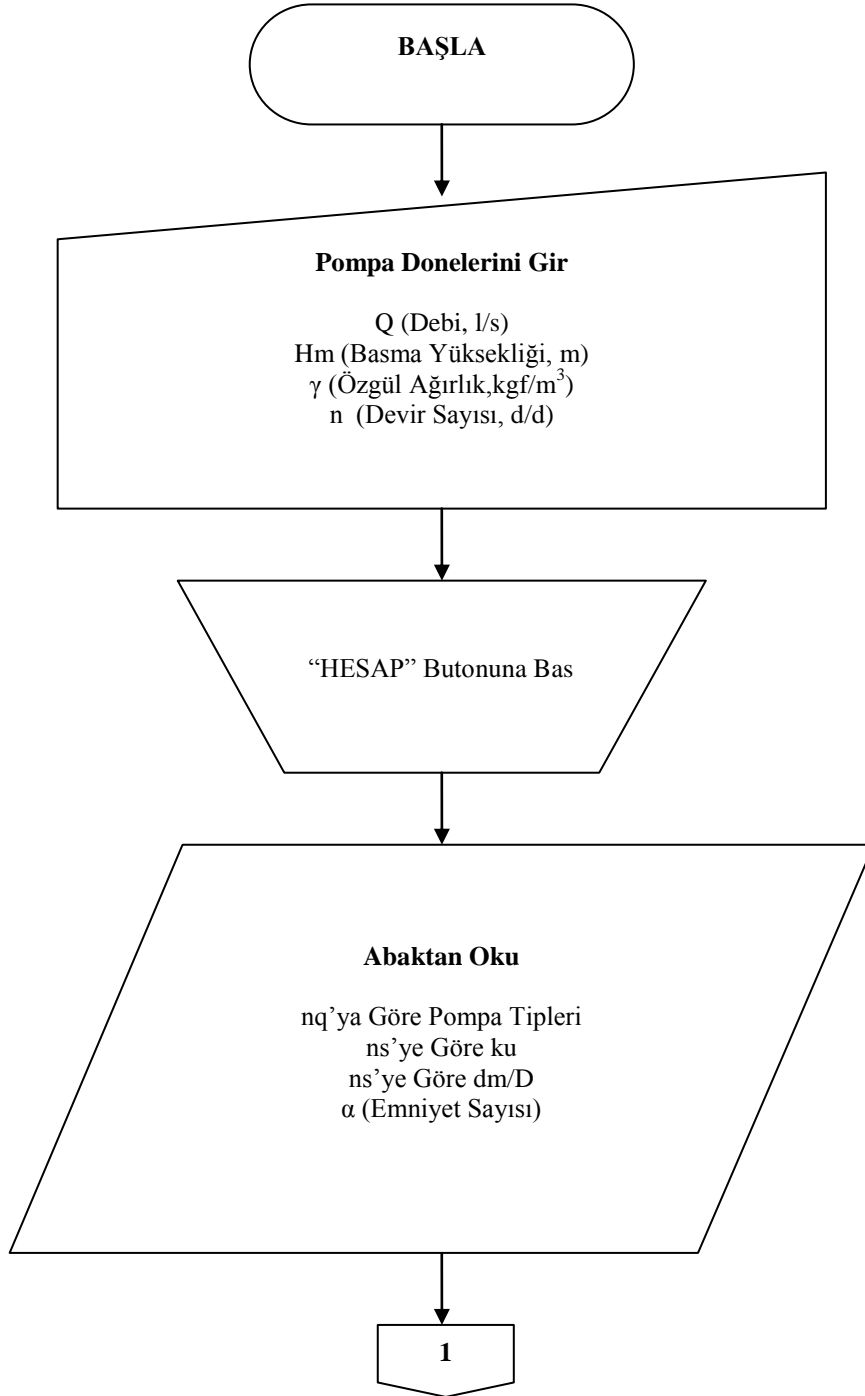
Şekil 5.4. Eksenel santrifij pompa kanat ana ölçüleri (Özgür,1972)

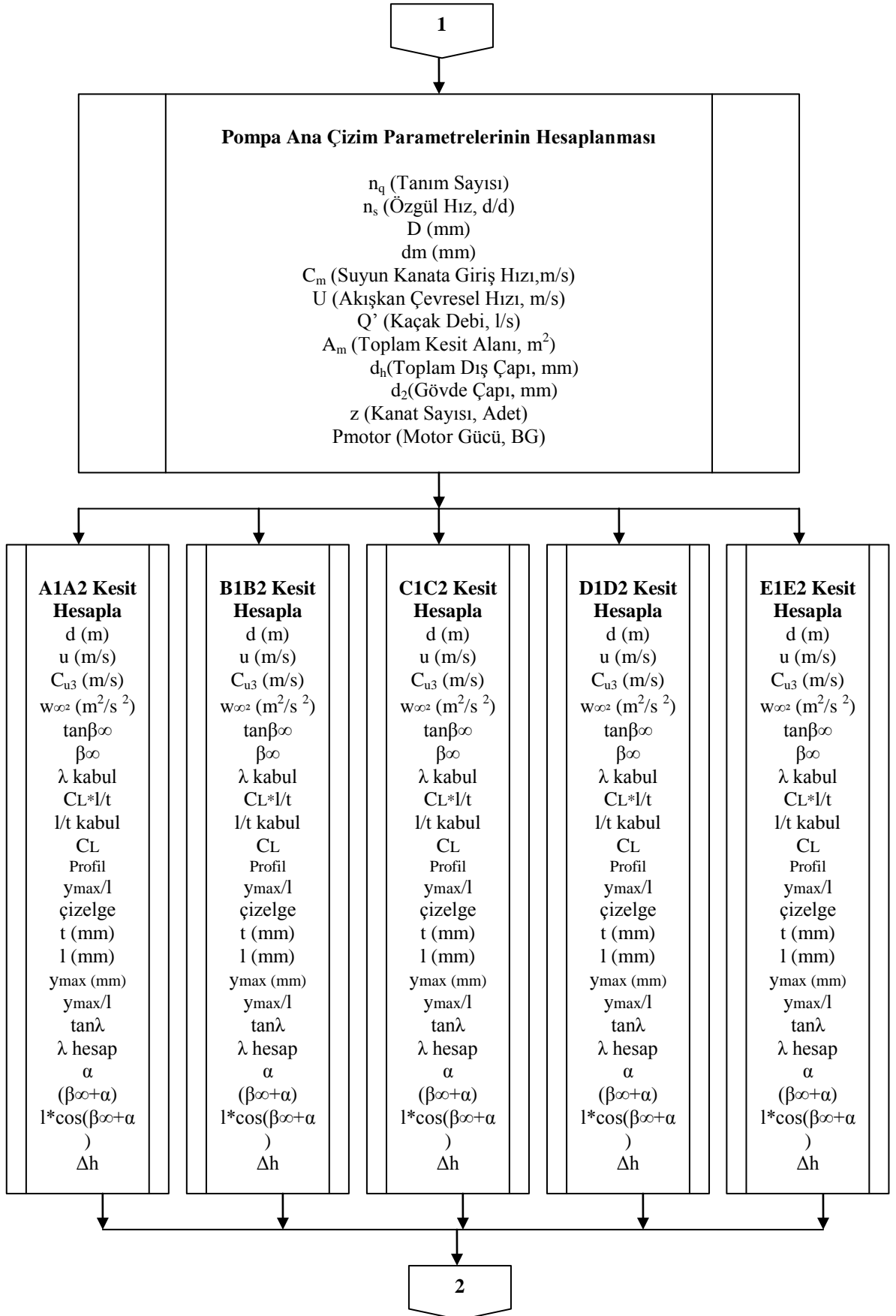
Çizelge 5.1. Aerodinamik profillerin ölçüleri (Trosklanski, 1982)

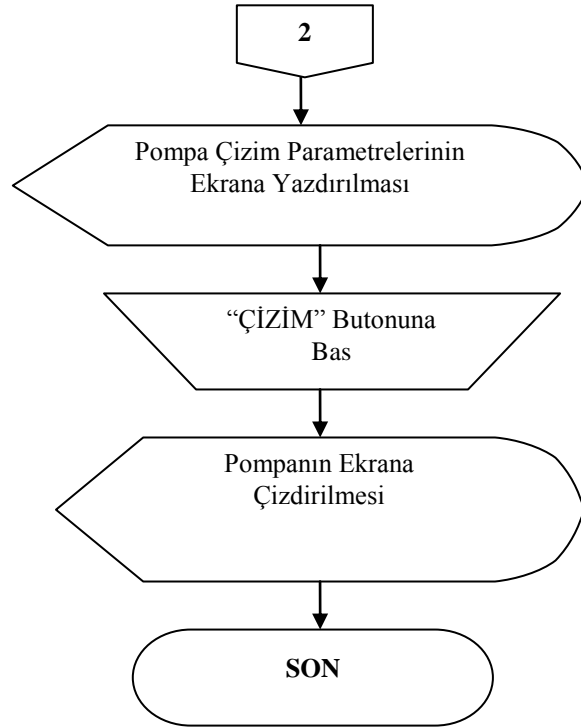
Profil No	x%	0	1.25	2.5	5.0	7.5	10	15	20	30	40	50	60	70	80	90	95	100
387	y	3.2	6.25	7.65	9.4	10.85	11.95	13.40	14.40	15.05	14.6	13.35	11.35	8.90	6.15	3.25	1.75	0.15
	yo	3.2	1.50	1.05	0.55	0.25	0.10	0.00	0.00	0.20	0.40	0.45	0.50	0.45	0.30	0.15	0.05	0.15
490	y	2.00	3.60	4.60	5.95	7.00	7.70	8.65	9.20	9.60	9.05	8.55	7.45	6.05	4.40	2.50	1.45	0.15
	yo	2.00	0.85	0.50	0.15	0.00	0.00	0.20	0.40	0.95	0.80	0.80	0.60	0.40	0.15	0.00	0.05	0.15
623	y	3.25	5.45	6.45	7.90	9.05	9.90	10.95	11.55	12.00	11.70	10.65	9.15	7.35	5.15	2.80	1.60	0.30
	yo	3.25	1.95	1.50	0.90	0.35	0.20	0.10	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
624	y	4.00	7.15	8.50	10.40	11.75	12.85	14.35	15.30	16.00	15.40	14.05	12.00	9.50	6.60	3.55	2.00	0.50
	yo	4.00	2.25	1.65	0.95	0.60	0.40	0.15	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Munk 6	y	0.00	1.98	2.81	4.03	4.94	5.71	6.82	7.55	8.22	8.05	7.26	6.03	4.58	3.06	1.55	0.88	0.00
	yo	0.00	-1.76	-2.20	-2.73	-3.03	-3.24	-3.47	-3.62	-3.70	-3.90	-3.94	-3.82	-3.48	-2.83	-1.77	-1.08	0.00
NACA 23012	y	0.00	2.67	3.61	4.91	5.80	6.43	7.19	7.50	7.55	7.14	6.41	5.47	4.36	3.08	1.68	0.92	0.00
	yo	0.00	-1.23	-1.71	-2.26	-2.61	-2.92	-3.50	-3.97	-4.46	-4.48	-4.17	-3.67	-3.00	-2.16	-1.23	-0.70	0.00
443	y	0.00	0.60	0.85	1.15	1.45	1.60	1.90	2.15	2.50	2.50	2.35	2.05	1.60	1.15	0.65	0.30	0.00
	yo	0.00	0.60	0.85	1.15	1.45	1.60	1.90	2.15	2.50	2.50	2.35	2.05	1.60	1.15	0.65	0.30	0.00

5.3. Eksenel Santrifüj Pompa Tasarım Programı Akış Şeması

5.3.1. Ana program

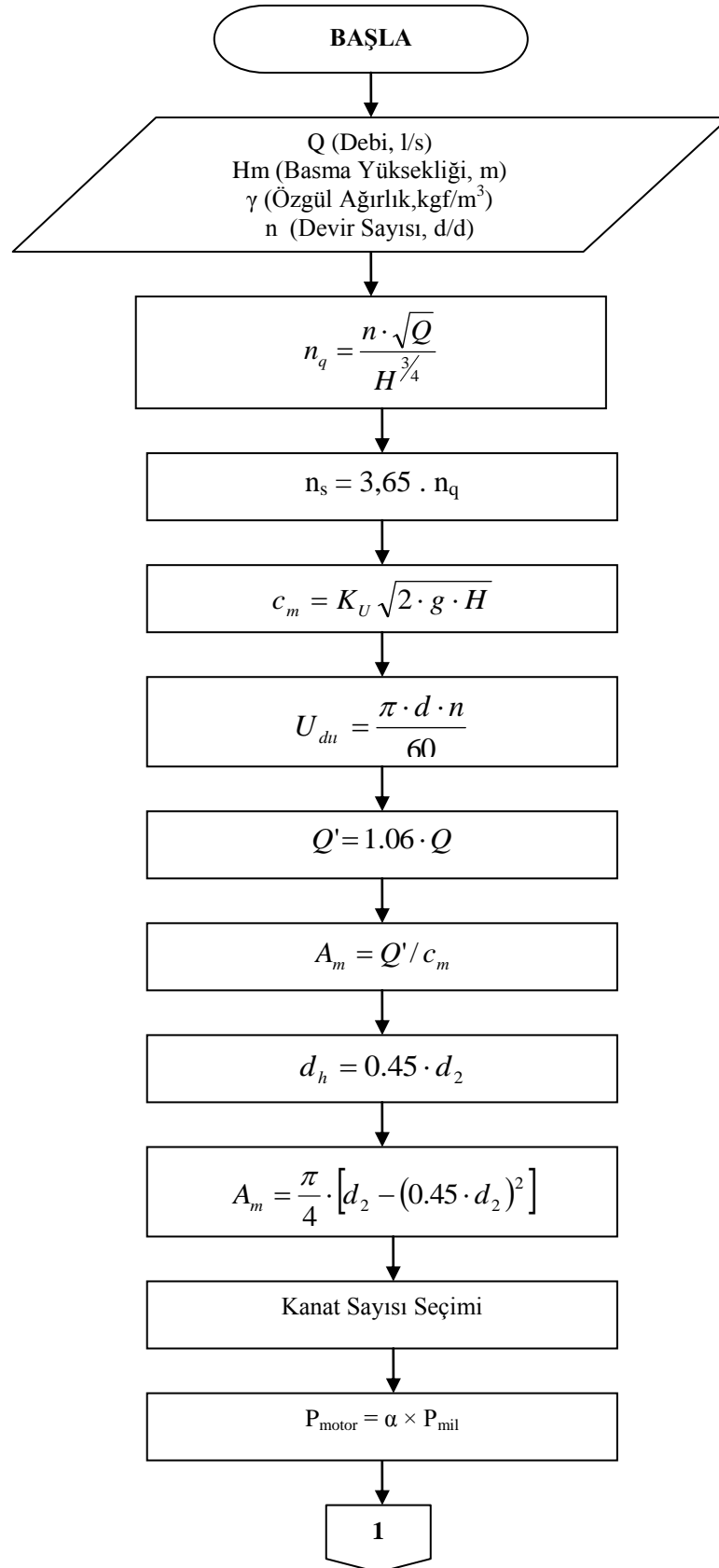


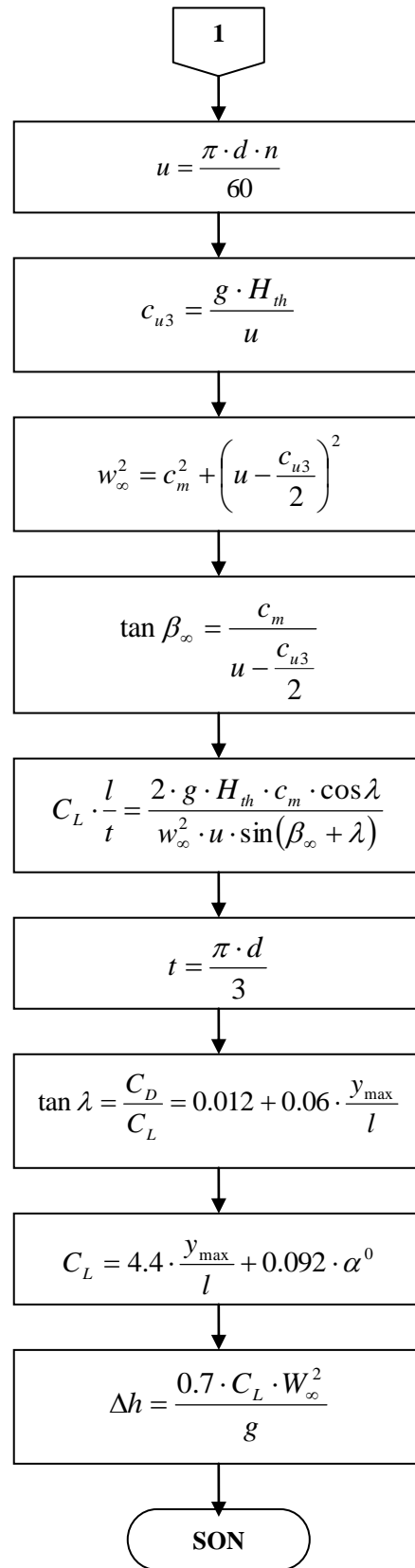




Şekil 5.5. Eksenel santrifüj pompaya ait ana programın akış şeması

5.3.2. Alt program





Şekil 5.6. Eksenel santrifuj pompaya ait alt programın akış şeması

5.4. Eksenel Santrifüj Pompa Tasarım Programının Uygulanması

Pompa doneleri:

$$Q=250l/s \text{ (Debi)}$$

$$Hm=4.5m \text{ (Basma Yüksekliği)}$$

$$n = 1450d / d \text{ (Devir Sayısı)}$$

$$\gamma = 1000 \text{ kgf/m}^3 \text{ (Özgül Ağırlık)}$$

5.4.1. Eksenel santrifüj pompa ile ilgili hesaplamalar

Özgül hız (n_s);

$$n_q = \frac{n\sqrt{Q}}{H^{3/4}} = \frac{1450\sqrt{0.25}}{4.5^{3/4}} = 237 d / d$$

$$n_s = 3,65 \times 237 = 710 d / d \text{ olarak bulunur.}$$

Bulunan değere göre, pompa çarkının tipi eksenel santrifüj çarktır.

D ve d_m çap tayini;

$n_s= 237 d/d$ için $K_U \approx 0.50$ olarak okunur.

$$c_m = K_U \sqrt{2 \cdot g \cdot H} = 0.50 \sqrt{2 \cdot g \cdot 4.5} = 4.7m/s$$

Kaçak debi;

$$Q' = 1.06 \cdot Q = 1.06 \cdot 250 = 265 l / s$$

Kesit alanı;

$$A_m = Q' / c_m = 0.265 / 4.7 = 0.0563m^3$$

d_h ve d_2 çapı;

$$d_h = 0.45 \cdot d_2$$

$$A_m = \frac{\pi}{4} \cdot [d_2 - (0.45 \cdot d_2)^2]$$

$$0.0563 = \frac{\pi}{4} \cdot [d_2 - (0.45 \cdot d_2)^2]$$

$$d_2 = 0.3m$$

$$d_h = 0.135m$$

Kanat sayısının tespiti;

Büyük özgül hızlar ve küçük manometrik yükseklikler için küçük kanat sayısı alındığı için kanat sayısını 3 olarak tespit ederiz.

Pompa mil gücü hesabı;

$$P_{mil} = \frac{\gamma \cdot Q \cdot H_m}{75 \cdot \eta_{pompa}} = \frac{1000 \cdot 0.25 \cdot 4.5}{75 \times 0.78} \approx 20BG$$

Motor gücü hesabı;

$$P_{motor} = \alpha \times P_{mil} = 1.25 \times 20 = 23BG$$

Çizelge 5.2. Kanat çiziminde kullanılan formüller

Sıra	Formül	Birim	Kesitler				
			A1A2	B1B2	C1C2	D1D2	E1E2
1	d	m	0.135	0.190	0.233	0.268	0.300
2	$u = \frac{\pi \cdot d \cdot n}{60}$	m/s	10.27	14.45	17.70	20.40	22.80
3	$c_{u3} = \frac{g \cdot H_{th}}{u}$	m/s	5.17	3.66	3.00	2.60	2.33
4	$w_{\infty}^2 = c_m^2 + \left(u - \frac{c_{u3}}{2}\right)^2$	m^2/s^2	78.5	179.0	276.0	385.0	467.0
5	$\tan \beta_{\infty} = \frac{c_m}{u - \frac{c_{u3}}{2}}$	-	0.574	0.349	0.272	0.231	0.208
6	β_{∞}	-	29°51'	19°16'	15°13'	13°00'	11°45'
7	λ (kabul)	-	1°	1°	1°	1°	1°
8	$C_L \cdot \frac{l}{t} = \frac{2 \cdot g \cdot H_{th} \cdot c_m \cdot \cos \lambda}{w_{\infty}^2 \cdot u \cdot \sin(\beta_{\infty} + \lambda)}$	-	1.128	0.522	0.333	0.246	0.198
9	$\frac{l}{t}$ (kabul)	-	0.900	0.830	0.770	0.715	0.670
10	$C_L = (C_L \cdot l/t) : l/t$	-	1.25	0.629	0.433	0.344	0.296
11	Profil (kabul)	-	387	490	490	490	490
12	$\frac{y_{\max}}{l}$ (kabul)	-	0.1505	0.01100	0.0850	0.0670	0.0600

Sıra	Formül	Birim	Kesitler				
			A1A2	B1B2	C1C2	D1D2	E1E2
13	$t = \frac{\pi \cdot d}{3}$	mm	141	199	244	280	314
14	$l = t \cdot l / t$	mm	127	166	188	200	210
15	$y_{\max} = l \cdot \frac{y_{\max}}{l}$	mm	19.1	18.2	16.0	14.0	12.6
16	y_{\max} (düzeltme sonrası)	mm	19.1	17.0	15.0	13.5	12.5
17	$\frac{y_{\max}}{l}$ (düzeltme sonrası)	-	0.1505	0.1050	0.0800	0.0670	0.0600
18	$\tan \lambda = \frac{C_D}{C_L} = 0.012 + 0.06 \cdot \frac{y_{\max}}{l}$	-	0.0210	0.0183	0.0168	0.0160	0.0156
19	λ (hesap)	-	1°12'	1°03'	0°58'	0°55'	0°53'
20	$\alpha^0 \Rightarrow C_L = 4.4 \cdot \frac{y_{\max}}{l} + 0.092 \cdot \alpha^0$	-	6°21'	2°00'	0°44'	0°32'	0°22'
21	$(\beta_{\infty} + \alpha^0)$	-	36°12'	21°16'	15°57'	13°32'	12°07'
22	$l \cdot \cos(\beta_{\infty} + \alpha^0)$	mm	103	155	180	194	204
23	$\Delta h = \frac{0.7 \cdot C_L \cdot W_{\infty}^2}{g}$	m	6.88	7.97	8.55	9.45	9.85

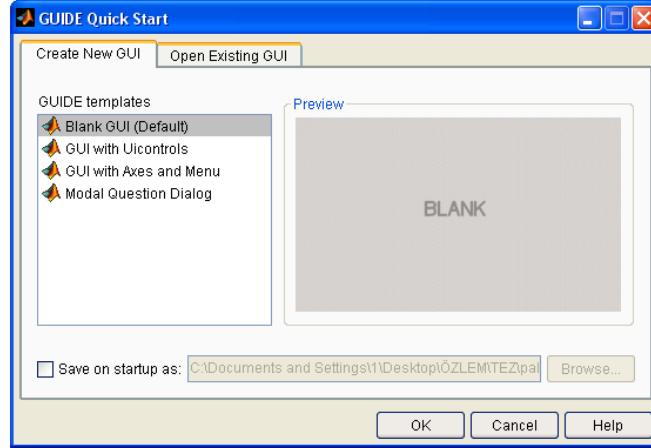
Bulunan değerlere ve hesaplanan açılara göre eşmerkezli olarak profiller gövde üzerine çizilir.

5.4.2. Aksel santrifüj pompa tasarım yazılımının oluşturulması

GUI'nin görüntülü kısmını oluşturmak:

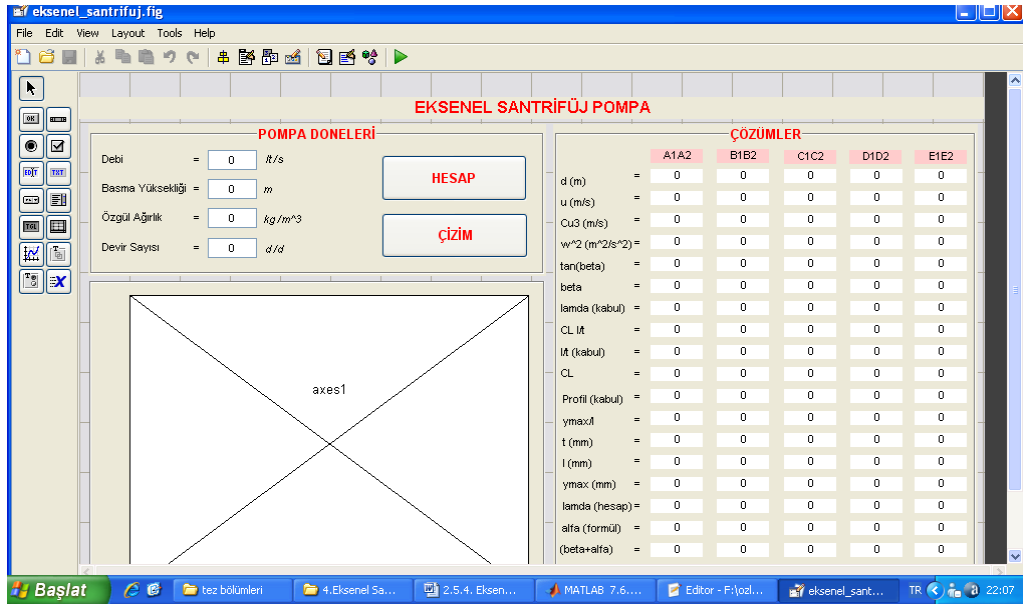
- Komut Satırına “guide” yazılır.

- Gelen ekrandan Blank GUI ikonu işaretlenir ve OK'ye basılır.



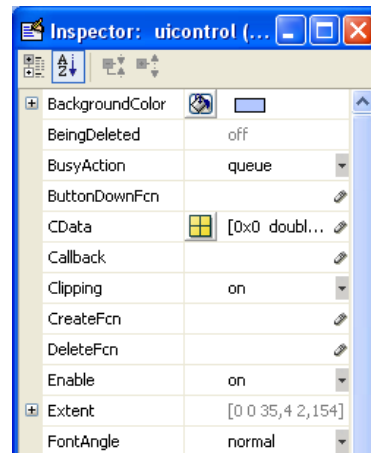
Şekil 5.7. “Guide Quick Start” ekranı

- GUI'nin nasıl görüneceğine karar verilir. Şöyle görünebilir:



Şekil 5.8. Eksenel santrifüj pompa GUI ön görünüm ekranı

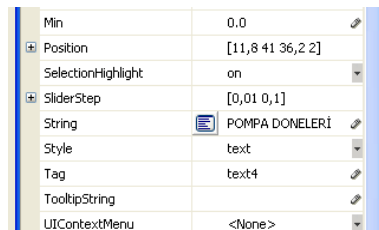
- Sol taraftaki Menüden StaticText, EditText, Axes ve PushButton'ları sürüklenip ekrana bırakılır. Yerlerine yerleştirilir.
- StaticText'lerden biri seçilip çift tıklanır. Aşağıdaki gibi bir ekranla karşılaşılır:



Şekil 5.9. GUI uicontrol ekranı

String kısmı “POMPA DONELERİ”ne dönüştürülür. İstenirse yazı boyutu, yazı kalınlığı gibi değerler de bu kısımdan değiştirilebilir.

Aynı işlem diğer StaticTextlere de uygulanır. String kısımları şekildeki gibi Debi, =, 0, m gibi değerlerle değiştirilir.



Şekil 5.10. GUI elemanlarında isimlendirme

- Çözümler bölümündeki sonuçların görüneceği StaticTextler'in String kısımlarına "0", Tag kısımlarına da sırasıyla şu değerler yazılır:

A1A2 Kesiti İçin:

answer_statictext_dA1A2

answer_statictext_uA1A2

answer_statictext_Cu3A1A2

answer_statictext_w2A1A2

answer_statictext_tanbetaA1A2

answer_statictext_betaA1A2

answer_statictext_lamdakabulA1A2

answer_statictext_CLltA1A2

answer_statictext_ltkabulA1A2

answer_statictext_CLA1A2

answer_statictext_ProfilkabulA1A2

answer_statictext_ymaxlA1A2

answer_statictext_tA1A2

answer_statictext_lA1A2

answer_statictext_ymaxA1A2

answer_statictext_lamdahesapA1A2

answer_statictext_alfaformulA1A2

answer_statictext_betaalfaA1A2

answer_statictext_betaalfaA1A2

answer_statictext_zA1A2

answer_statictext_NmA1A2

A1A2 ismi B1B2, C1C2, D1D2, E1E2 olarak değiştirilerek diğer kesitler için de benzer etiketler oluşturulur.

- EditText'lerden birine çift tıklanıp String kısmına "0", Tag kısmına debi_editText yazılır.

- Diğer EditText'lerin String kısmına da "0" yazılır ancak Tag kısımlarına sırasıyla şu deęerler yazılır:

basmayuksekligi_editText

devirsayisi_editText

ozgulagirlik_editText

- PushButton'lardan birine çift tıklanıp String kısmına "HESAP", Tag kısmına hesap_pushbutton yazılır.

- Diğer PushButton'un String kısımlarına da "ÇİZİM", Tag kısmına cizim_pushbutton yazılır.

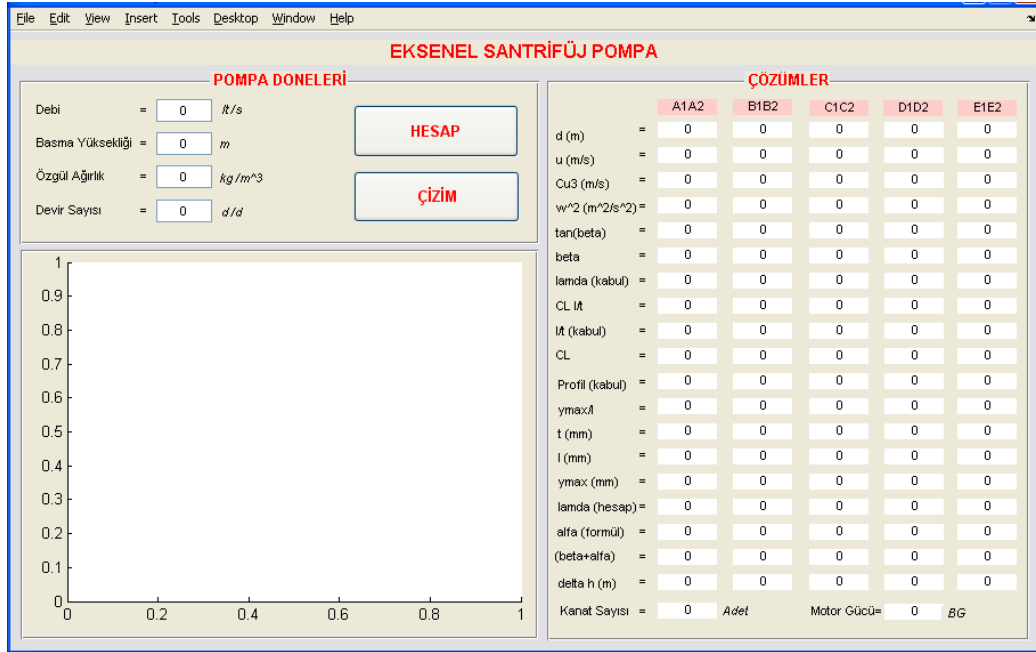
- Axes'e çift tıklanıp Tag kısmına axes1 yazılır.

- Sayfa "eksenel_santrifuj" olarak kaydedilir. eksenel_santrifuj.fig ve eksenel_santrifuj .m olmak üzere iki dosya oluşur. Oluşan m.file'ı açılır.

GUI'ye kod yazmak:

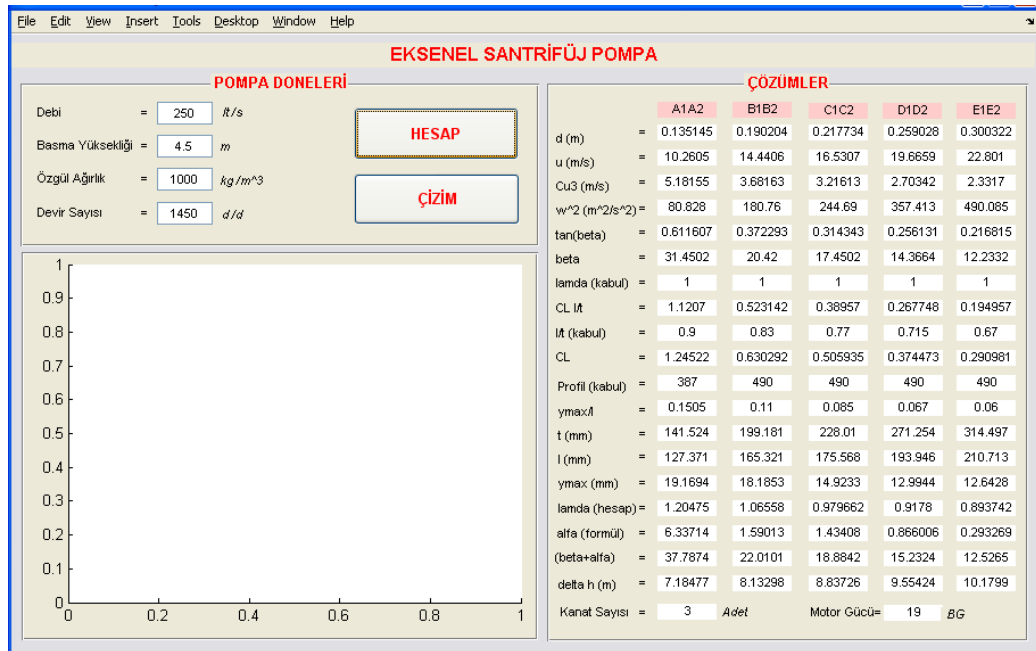
eksenel_santrifuj.m isimli m.file dosyasına Ekler'de verilen kodları yazarız.

Run  ikonuna tıkladığımızda Şekil 5.11'deki görünümü elde ederiz:



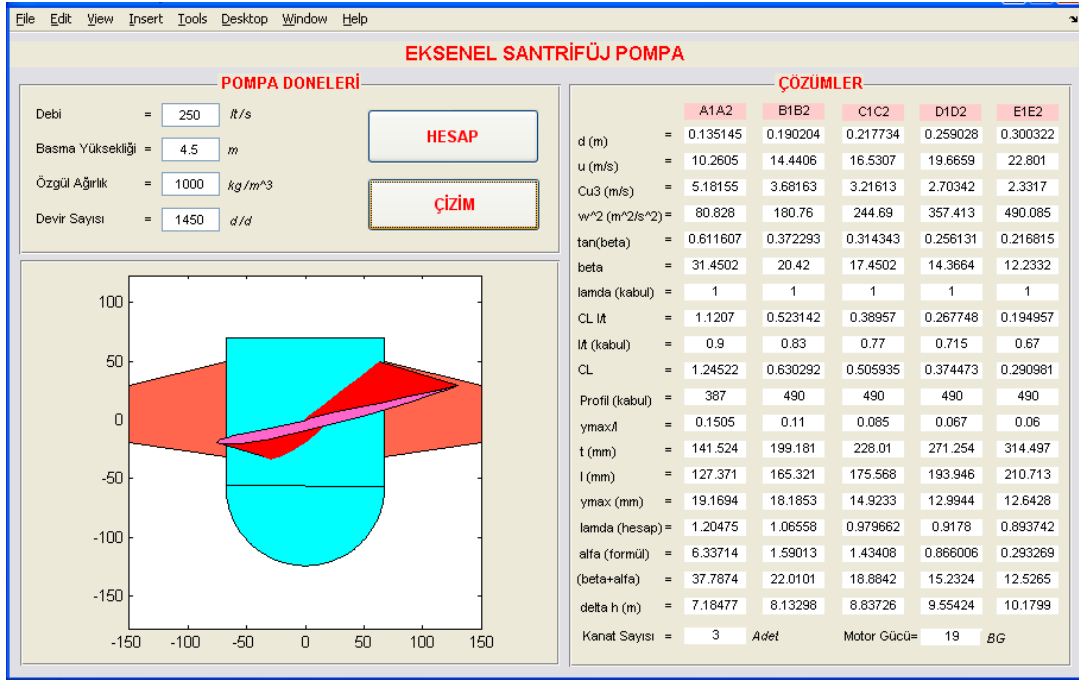
Şekil 5.11. Eksenel santrifüj pompa hesap ve çizim ekranı

Sol taraftaki pompa donelerini girerek “HESAP” butonuna bastığımızda MATLAB hesapları yapar ve sonuçlar ekranda görüntülenir.



Şekil 5.12. Eksenel santrifüj pompa ekranında “HESAP” butonuna basılmış durum

“ÇİZİM” butonuna bastığımızda ana elemanların çizimi axes bölümünde görüntülenir.



Şekil 5.13. Eksenel santrifüj pompa ekranında “ÇİZİM” butonuna basılmış durum

Program MATLAB Kurulu olmayan bilgisayarda bu haliyle çalışmaz. Çalışmasını sağlamak için programı .exe uzantılı hale dönüştürmemiz gerekir. Bunun için m.file açıkken komut penceresine şu komutu yazarız:

```
>>mcc -m eksenel_santrifuj
```

Lcc compiler seçeneğini seçtiğimizde program .exe uzantılı hale gelir.

BÖLÜM 6

SONUÇ VE ÖNERİLER

Geleneksel yöntemlerle tasarım yapılmak istenildiğinde her farklı pompa dizaynı için uzun bir hesaplama süreci ile birlikte pompanın imalatı için gerekli dizayn resimlerinin hazırlanması zaman kaybına neden olmaktadır. Bu çalışmanın yapılmasının nedeni mevcut programların yüksek maliyetli olmasıdır. Bunun yanında tasarlanan pompanın hesaplama yöntemine bir iyileştirme yapılmak istenildiğinde bunun paket program ile çok zor olacağı kaçınılmazdır.

Gerçekleştirilen bu çalışmada paletli, dişli, radyal santrifüj ve eksenel santrifüj pompa tasarım hesaplamalarını yaparak bu pompalara ait ana elemanların resimlerini ölçekli olarak çizecek açık kaynak kodlu bir yazılımın gerçekleştirilmesi amaçlanmıştır.

Pompa tasarım yazılımı gerçekleştirilmeden önce ilk olarak programların akış diyagramları oluşturulmuş ve böylece yazılımın hangi parametrelere ihtiyaç duyacağı ve nasıl çalışacağı belirlenmiştir. Bunun ardından yazılım gerçekleştirilmiştir. Çıkarılan bu akış diyagramları son kullanıcıya gerçekleştirilen bu yazılımın çalışma yöntemini açık bir şekilde ifade etmektedir. Bunun yanında gerçekleştirilen yazılımın açık kaynak kodlu bir yazılım olması sebebiyle ileride programa müdahale edilmek istenildiği zaman çıkarılan bu akış diyagramları sayesinde programın çalışma mekanizmasının anlaşılması kolaylaştırılmıştır.

Pompa tasarımında formüllerle hesaplanabilen parametrelerin yanında grafiklerden okunarak hesaplamalara dahil edilen çeşitli parametreler de mevcuttur. Bu nedenle yazılımın gerçekleştirileceği platformun belirlenmesinde bu tip hesaplamalarda çok esnek bir program olan Matlab yazılımı tercih edilmiştir. Gerçekleştirilen yazılımların tasarımcılar tarafından daha fonksiyonel olarak kullanılabilmesi için görsel arayüzler oluşturulmuş ve böylece programın kullanımı kolaylaştırılmıştır.

Paletli, dişli, radyal santrifüj ve eksenel santrifüj pompa tasarımı hesaplamalarını yapacak yazılımları test etmek amacıyla her pompa için karşılaştırma örneği olarak elle tasarım hesapları gerçekleştirilmiş ve gerçekleştirilen yazılım aynı parametrelerle çalıştırılarak programdan elde edilen sonuçlar elle hesaplanarak elde edilen sonuçlarla karşılaştırılmış ve gerçekleştirilen pompa tasarımı hesaplama yazılımlarının doğru çalıştıkları gösterilmiştir.

Hazırlanan yazılımda karşılaşılan en büyük problem tasarım parametreleri hesaplatılan pompaların çizilmesini sağlayacak yazılımın gerçekleştirilmesinde yaşanmıştır. Başlangıçta Matlab programı ile bir katı model çizim programı olan SolidWorks programı haberleştirilerek tasarlanan parametrelere bağlı olarak ana eleman çizimlerinin doğrudan katı model programı üzerinden çizdirilmesi amaçlanmış ancak bu katı model programı ile Matlab senkronize olarak çalıştırılamamıştır. Pompa ana elemanlarının çizdirilmesi Matlab yazılımının grafik komutları yardımıyla gerçekleştirilmiştir. Gelecek çalışmalarda gerçekleştirilen bu yazılım ile bir katı model programı senkronize olarak çalıştırılarak pompa elemanlarının üç boyutlu olarak çizdirilmesi sağlanabilir.

Yazılımın üstünlüklerinden birisi de aynı tasarım parametreleri için paletli, dişli, radyal santrifüj ve eksenel santrifüj pompa olmak üzere dört farklı tipte pompa için tasarım hesaplamalarının dakikalar içerisinde yapılarak en uygun pompaya karar verilmesi aşamasında tasarımcıya büyük bir avantaj sağlamasıdır.

KAYNAKLAR DİZİNİ

Exner, H., Freitag, R., Geis, H., Lang, R., Oppolzer, J., Schwab P., Sumpf E., Ostendorff, U. and Mannesman Rexroth AG, 1998, Rexroth Hydraulics , Germany

Karmel, A.M., Modelling and Analysis of The Dynamics of a Variable-Displacement Vane Pump, Volume: 110, Issue: 2

Özgür, C., 1972, Su makinaları dersleri, Çağlayan Basımevi, İstanbul, 228-242

Pancar, Y., 1980, Santrifüj pompalar ders notları, Eskişehir

Pancar, Y., 2007, Hidrolik makineler ders notları, Eskişehir

Yalçın, K., 1998, Hacimsal ve santrifüj pompalar, Çağlayan Kitabevi, İstanbul, 1 s.

Trosklanski, A.T., 1982, Impeller pumps, Pergamon Press, Newyork, 228-258

MEGEP Plastik Teknolojisi, 2007, Hidrolik Devre, Ankara

<http://www.dengeteknoloji.com/atik-su-pompalari.asp>

<http://extreme.infomagic.net/static/SmGerotor.jpeg>

EKLER

Ek.1. Paletli Pompa Tasarım Programı

Ek.2. Dişli Pompa Tasarım Programı

Ek.3. Radyal Santrifüj Pompa Tasarım Programı

Ek.4. Eksenel Santrifüj Pompa Tasarım Programı

Ek.1. Paletli Pompa Tasarım Programı

```
function varargout = paletli(varargin)
% PALETTLI M-file for paletli.fig
%     PALETTLI, by itself, creates a new PALETTLI or raises the
existing
%     singleton*.
%
%     H = PALETTLI returns the handle to a new PALETTLI or the handle
to
%     the existing singleton*.
%
%     PALETTLI('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in PALETTLI.M with the given input
arguments.
%
%     PALETTLI('Property','Value',...) creates a new PALETTLI or raises
the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before paletli_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to paletli_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help paletli

% Last Modified by GUIDE v2.5 24-Nov-2009 22:35:51
global R
global Dr
global t
global deb
global Dmil
global Dgobek
global l
global z
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @paletli_OpeningFcn, ...
                  'gui_OutputFcn',  @paletli_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before paletli is made visible.
function paletli_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to paletli (see VARARGIN)

% Choose default command line output for paletli
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes paletli wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = paletli_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function basinc_editText_Callback(hObject, eventdata, handles)
% hObject    handle to basinc_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of basinc_editText as
text
%         str2double(get(hObject,'String')) returns contents of
basinc_editText as a double
%store the contents of basinc_editText as a string. if the string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default basinc_editText to
zero
if (isempty(input))
    set(hObject,'String','0')
end
end

```



```

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function basinc_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to basinc_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function malzemegerilme_editText_Callback(hObject, eventdata, handles)
% hObject    handle to malzemegerilme_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of
malzemegerilme_editText as text
%       str2double(get(hObject, 'String')) returns contents of
malzemegerilme_editText as a double
%store the contents of malzemegerilme_editText as a string. if the
string
%is not a number then input will be empty

input = str2num(get(hObject, 'String'));

%checks to see if input is empty. if so, default
malzemegerilme_editText to zero
if (isempty(input))
    set(hObject, 'String', '0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function malzemegerilme_editText_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to malzemegerilme_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
function discap_editText_Callback(hObject, eventdata, handles)
% hObject    handle to discap_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of discap_editText as
text
%         str2double(get(hObject,'String')) returns contents of
discap_editText as a double
%store the contents of discap_editText as a string. if the string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default discap_editText to
zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function discap_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to discap_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function devirsayisi_editText_Callback(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
devirsayisi_editText as text
%         str2double(get(hObject,'String')) returns contents of
devirsayisi_editText as a double
%store the contents of devirsayisi_editText as a string. if the string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default devirsayisi_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function devirsayisi_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function paletsayisi_editText_Callback(hObject, eventdata, handles)
% hObject      handle to paletsayisi_editText (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
paletsayisi_editText as text
%       str2double(get(hObject,'String')) returns contents of
paletsayisi_editText as a double
%store the contents of paletsayisi_editText as a string. if the string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default paletsayisi_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function paletsayisi_editText_CreateFcn(hObject, eventdata, handles)
% hObject      handle to paletsayisi_editText (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function kanatyuksekligi_editText_Callback(hObject, eventdata,
handles)
% hObject      handle to kanatyuksekligi_editText (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
kanatyuksekligi_editText as text
%       str2double(get(hObject,'String')) returns contents of
kanatyuksekligi_editText as a double

```

```

%store the contents of kanatyuksekligi_editText as a string. if the
string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default
kanatyuksekligi_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function kanatyuksekligi_editText_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to kanatyuksekligi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ozgulagirlik_editText_Callback(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
ozgulagirlik_editText as text
%         str2double(get(hObject,'String')) returns contents of
ozgulagirlik_editText as a double
%store the contents of ozgulagirlik_editText as a string. if the
string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default ozgulagirlik_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function ozgulagirlik_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function genelverim_editText_Callback(hObject, eventdata, handles)
% hObject     handle to genelverim_editText (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of genelverim_editText
as text
%     str2double(get(hObject,'String')) returns contents of
genelverim_editText as a double
%store the contents of genelverim_editText as a string. if the string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default genelverim_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function genelverim_editText_CreateFcn(hObject, eventdata, handles)
% hObject     handle to genelverim_editText (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in hesap_pushbutton.
function hesap_pushbutton_Callback(hObject, eventdata, handles)
global R
global Dr
global t
global deb
global Dmil
global Dgobek
global l
global z
% hObject     handle to hesap_pushbutton (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

```

```

% --- Executes on button press in add_pushbutton.
P = get(handles.basinc_editText, 'String');
sigma = get(handles.malzemegerilme_editText, 'String');
R = get(handles.discap_editText, 'String');
n = get(handles.devirsayisi_editText, 'String');
z = get(handles.paletsayisi_editText, 'String');
b = get(handles.kanatyuksekligi_editText, 'String');
ng = get(handles.genelverim_editText, 'String');
gamma = get(handles.ozgulagirlik_editText, 'String');
% P, Sigma, R, n, z, b and gamma are variables of Strings type, and need to
be converted
% to variables of Number type before they can be added together

functionH = str2num(P)*1000/str2num(gamma);
H = num2str(functionH);
% need to convert the answer back into String type to display it
set(handles.answer_basmayuksekligi, 'String', H);
guidata(hObject, handles);

functionDr = str2num(R)*0.76 ;
Dr = num2str(functionDr);
% need to convert the answer back into String type to display it
set(handles.answer_rotorcapi, 'String', Dr);
guidata(hObject, handles);

functionl = str2num(R)- str2num(Dr)+3;
l = num2str(functionl);
% need to convert the answer back into String type to display it
set(handles.answer_paletboyu, 'String', l);
guidata(hObject, handles);
functionVb =
pi*str2num(R)*1/1000*str2num(R)*1/1000/4*str2num(b)*1/1000;
Vb = num2str(functionVb);
% need to convert the answer back into String type to display it
set(handles.answer_butunhacim, 'String', Vb);
guidata(hObject, handles);

functionVr =
pi*str2num(Dr)*1/1000*str2num(Dr)*1/1000/4*str2num(b)*1/1000;
Vr = num2str(functionVr);
% need to convert the answer back into String type to display it
set(handles.answer_rotorhacmi, 'String', Vr);
guidata(hObject, handles);

functionQ = (str2num(Vb)-str2num(Vr))*str2num(n)/60;
Q = num2str(functionQ);
% need to convert the answer back into String type to display it
set(handles.answer_debi, 'String', Q);
guidata(hObject, handles);

functionNe =
ceil(str2num(gamma)*str2num(Q)*str2num(H)/75/str2num(ng)*100);
Ne = num2str(functionNe);
% need to convert the answer back into String type to display it
set(handles.answer_verim, 'String', Ne);
guidata(hObject, handles);

```

```

%standart mil caplari 15-20-25-30-35-40-50-60-70-80-90
milcapi = (14.4*(str2num(Ne))^(1/3)/(str2num(n))^(1/3)*10);
if milcapi<=15;
    functionDmil =15;
elseif 15<milcapi<=20;
    functionDmil =20;
elseif 20<milcapi<=25;
    functionDmil =25;
elseif 25<milcapi<=30;
    functionDmil =30;
elseif 30<milcapi<=35;
    functionDmil =35;
elseif 35<milcapi<=40;
    functionDmil =40;
elseif 40<milcapi<=50;
    functionDmil =50;
elseif 50<milcapi<=60;
    functionDmil =60;
elseif 60<milcapi<=70;
    functionDmil =70;
elseif 70<milcapi<=80;
    functionDmil =80;
elseif 80<milcapi<=90;
    functionDmil =90;
end
functionDmil;
Dmil = num2str(functionDmil);
% need to convert the answer back into String type to display it
set(handles.answer_milcapi, 'String', Dmil);
guidata(hObject, handles);

functionDgobek = ceil(str2num(Dmil)+3);
Dgobek = num2str(functionDgobek);
% need to convert the answer back into String type to display it
set(handles.answer_gobekcapi, 'String', Dgobek);
guidata(hObject, handles);

functiont = ceil(str2num(P)*str2num(R)/2/str2num(sigma));
t = num2str(functiont);
% need to convert the answer back into String type to display it
set(handles.answer_etkalinligi, 'String', t);
guidata(hObject, handles);

if str2num(R) <= 100
    functiondeb = 30;
else
    functiondeb = 20;
end
functiondeb;
deb = num2str(functiondeb);
% need to convert the answer back into String type to display it
set(handles.answer_emmebasma, 'String', deb);
guidata(hObject, handles);

```

```

% --- Executes on button press in anaelemancizim_pushputton.
function anaelemancizim_pushputton_Callback(hObject, eventdata,
handles)
% hObject    handle to anaelemancizim_pushputton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%selects axes1 as the current axes, so that
%Matlab knows where to plot the data
axes(handles.axes1)

%axis([-200 200 -200 200])
style='-';

%H=circle(center,radius,theta1,theta2,alpha,style);

global R
global Dr
global t
global deb
global Dmil
global Dgobek
global l
global z

R=str2num(R);
Dr=str2num(Dr);
t=str2num(t);
deb=str2num(deb);
Dmil=str2num(Dmil);
Dgobek=str2num(Dgobek);
l=str2num(l);
z=str2num(z);

Dem=deb;
dmil=Dmil;
dgobek=Dgobek;
PaletBoyu=l;
PaletKalinligi=3;
PaletSayisi=z;

%-----

%Pompa Haznesinin Çizimi

%Haznenin Dış Sınırlarının Çizimi

x0=0; y0=Dr/2-R/2;
x2=-sqrt((R/2+t)^2-(Dem/2+t)^2); y2=y0-Dem/2-t;
cizim(1,:)=[x2 y2];

center=[x0 y0];
baslamaAci=atan2(-Dem/2-t,-sqrt((R/2+t)^2-(Dem/2+t)^2))*180/pi+360;
bitisAci=atan2(-Dem/2-t,sqrt((R/2+t)^2-(Dem/2+t)^2))*180/pi+360;

```



```

H=circle1(center,R/2+t,baslamaAci+1,bitisAci,0,style);
H=H'; s1=size(H);
cizim(2:s1(1,1)+1,:)=H(1:s1(1,1),:);

x3=-x2; y3=y2;
cizim(s1(1,1)+2,:)= [x3 y3];

x4=x3+20; y4=y2;
cizim(s1(1,1)+3,:)= [x4 y4];

x5=x4; y5=y0+Dem/2+t;
cizim(s1(1,1)+4,:)= [x5 y5];

x6=x3; y6=y5;
cizim(s1(1,1)+5,:)= [x6 y6];

baslamaAci=atan2(Dem/2+t,sqrt((R/2+t)^2-(Dem/2+t)^2))*180/pi;
bitisAci=atan2(Dem/2+t,-sqrt((R/2+t)^2-(Dem/2+t)^2))*180/pi;
H=circle1(center,R/2+t,baslamaAci+1,bitisAci,0,style);
H=H'; s2=size(H);
cizim(s1(1,1)+6:s1(1,1)+5+s2(1,1),:)=H(1:s2(1,1),:);

x7=x2; y7=y5;
cizim(s1(1,1)+s2(1,1)+6,:)= [x7 y7];

x8=x2-20; y8=y5;
cizim(s1(1,1)+s2(1,1)+7,:)= [x8 y8];

x1=x8; y1=y2;
cizim(s1(1,1)+s2(1,1)+8,:)= [x1 y1];

fill(cizim(:,1),cizim(:,2),'m')
%-----

%Haznenin İç Sınırlarının Çizimi

clear cizim
cizim=[x1 y0-Dem/2 ; x4 y0-Dem/2 ; x5 y0+Dem/2 ; x8 y0+Dem/2];
hold on
fill(cizim(:,1),cizim(:,2),[0.8 0.8 0.8])
hold off
clear cizim
H=circle1(center,R/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),[0.8 0.8 0.8])
hold off
%-----

%Rotor Çizimi

clear cizim
center=[0 0];
H=circle1(center,Dr/2,1,360,0,style);
cizim=H';

```

```

hold on
fill(cizim(:,1),cizim(:,2),'c')
hold off
%-----

%Göbek Çizimi

H=circle1(center,dgobek/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'y')
hold off
%-----

%Mil Çizimi

H=circle1(center,dmil/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'b')
hold off
%-----

%Palet Çizimi

for i=0:PaletSayisi-1
    clear cizim

    center=[0 Dr/2-PaletBoyu+PaletKalinligi/2];

H=circle1(center,PaletKalinligi/2,180,360,i*360/PaletSayisi,style);
H=H'; s1=size(H);
cizim(1:s1(1,1),:)=H(1:s1(1,1),:);

    alpha=i*360/PaletSayisi;
    cizim(s1(1,1)+1,:)= [cosd(alpha) -sind(alpha); sind(alpha)
cosd(alpha)]*[PaletKalinligi/2;Dr/2-PaletBoyu+PaletKalinligi/2];
    cizim(s1(1,1)+2,:)= [cosd(alpha) -sind(alpha); sind(alpha)
cosd(alpha)]*[PaletKalinligi/2;Dr/2];
    cizim(s1(1,1)+3,:)= [cosd(alpha) -sind(alpha); sind(alpha)
cosd(alpha)]*[-PaletKalinligi/2;Dr/2];
    cizim(s1(1,1)+4,:)= [cosd(alpha) -sind(alpha); sind(alpha)
cosd(alpha)]*[-PaletKalinligi/2;Dr/2-PaletBoyu+PaletKalinligi/2];
    hold on
    fill(cizim(:,1),cizim(:,2),'r')
    hold off
end
%-----
set(gca,'DataAspectRatio',[1 1 1], 'PlotBoxAspectRatio',[1 1 1])

guidata(hObject, handles); %updates the handles

function P=circle1(center,radius,theta1,theta2,alpha,style)
%-----

```

```

% H=CIRCLE(CENTER,RADIUS,NOP,STYLE)
% This routine draws a circle with center defined as
% a vector CENTER, radius as a scalar RADIS. NOP is
% the number of points on the circle. As to STYLE,
% use it the same way as you use the routine PLOT.
% Since the handle of the object is returned, you
% use routine SET to get the best result.
%
% Usage Examples,
%
% circle([1,3],3,1000,':');
% circle([2,4],2,1000,'--');
%
% Zhenhai Wang <zhenhai@ieee.org>
% Version 1.00
% December, 2002
%-----
NOP=360;
if (nargin <3),
    error('Please see help for INPUT DATA.');
```

```
elseif (nargin==3)
    style='b-';
end;
THETA=linspace(0,2*pi,NOP);
RHO=ones(1,NOP)*radius;
[X,Y] = pol2cart(THETA,RHO);
X=X+center(1);
Y=Y+center(2);
X1=X(thetal:theta2);
Y1=Y(thetal:theta2);
P=[cosd(alpha) -sind(alpha); sind(alpha) cosd(alpha)]*[X1;Y1];
% hold on
% H=plot(P(1,:),P(2,:),style);
% axis square;
% hold off

```

Ek.2. Dişli Pompa Tasarım Programı

```
function varargout = disli(varargin)
% DISLI M-file for disli.fig
%     DISLI, by itself, creates a new DISLI or raises the existing
%     singleton*.
%
%     H = DISLI returns the handle to a new DISLI or the handle to
%     the existing singleton*.
%
%     DISLI('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in DISLI.M with the given input
arguments.
%
%     DISLI('Property','Value',...) creates a new DISLI or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before disli_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to disli_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help disli

% Last Modified by GUIDE v2.5 24-May-2009 03:32:22

global da
global dt
global dg
global df
global m
global t
global r1
global r2
global z
global Dem
global et
global dmil
global dgobek

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @disli_OpeningFcn, ...
                  'gui_OutputFcn',  @disli_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
```

```

        gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before disli is made visible.
function disli_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to disli (see VARARGIN)

% Choose default command line output for disli
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes disli wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = disli_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function devirsayisi_editText_Callback(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
devirsayisi_editText as text
%        str2double(get(hObject,'String')) returns contents of
devirsayisi_editText as a double
%store the contents of devirsayisi_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default devirsayisi_editText
to zero
if (isempty(input))

```

```

        set(hObject, 'String', '0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function devirsayisi_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function basmayuksekligi_editText_Callback(hObject, eventdata,
handles)
% hObject    handle to basmayuksekligi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of
basmayuksekligi_editText as text
%         str2double(get(hObject, 'String')) returns contents of
basmayuksekligi_editText as a double
%store the contents of basmayuksekligi_editText as a string. if the
string
%is not a number then input will be empty
input = str2num(get(hObject, 'String'));

%checks to see if input is empty. if so, default
basmayuksekligi_editText to zero
if (isempty(input))
    set(hObject, 'String', '0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function basmayuksekligi_editText_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to basmayuksekligi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

function ozgulagirlik_editText_Callback(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
ozgulagirlik_editText as text
%         str2double(get(hObject,'String')) returns contents of
ozgulagirlik_editText as a double
%store the contents of ozgulagirlik_editText as a string. if the
string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default ozgulagirlik_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ozgulagirlik_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dissayisi_editText_Callback(hObject, eventdata, handles)
% hObject    handle to dissayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dissayisi_editText
as text
%         str2double(get(hObject,'String')) returns contents of
dissayisi_editText as a double
%store the contents of dissayisi_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default dissayisi_editText to
zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

```

```

% --- Executes during object creation, after setting all properties.
function dissayisi_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dissayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function modul_editText_Callback(hObject, eventdata, handles)
% hObject    handle to modul_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of modul_editText as
text
%       str2double(get(hObject,'String')) returns contents of
modul_editText as a double
%store the contents of modul_editText as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));

```

```

%checks to see if input is empty. if so, default modul_editText to
zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

```

```

% --- Executes during object creation, after setting all properties.
function modul_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to modul_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in hesap_pushbutton.
function hesap_pushbutton_Callback(hObject, eventdata, handles)
global da
global dt
global dg
global df

```



```

global m
global t
global r1
global r2
global z

% hObject    handle to hesap_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
n = get(handles.devirsayisi_editText, 'String');
H = get(handles.basmayuksekligi_editText, 'String');
gama = get(handles.ozgulagirlik_editText, 'String');
z = get(handles.dissayisi_editText, 'String');
m = get(handles.modul_editText, 'String');
b = 30 %pompayuksekligi
alfa = 20 %kanatacisi
% n,H,gama,z,m,b and alfa are variables of Strings type, and need to
be converted
% to variables of Number type before they can be added together

functiont = str2num(m)*pi;
t = num2str(functiont);
% need to convert the answer back into String type to display it
set(handles.answer_dishatvesi_staticText, 'String', t);
guidata(hObject, handles);

functionh = 2.166*str2num(m);
h = num2str(functionh);
% need to convert the answer back into String type to display it
set(handles.answer_disyuksekligi_staticText, 'String', h);
guidata(hObject, handles);

functiondt = str2num(m)*str2num(z);
dt = num2str(functiondt);
% need to convert the answer back into String type to display it
set(handles.answer_taksimatdaairesicapi_staticText, 'String', dt);
guidata(hObject, handles);

functionda = str2num(m)*(str2num(z)+2);
da = num2str(functionda);
% need to convert the answer back into String type to display it
set(handles.answer_disustudairesicapi_staticText, 'String', da);
guidata(hObject, handles);

functiondg = str2num(dt)*cos(alfa*3.141/180);
dg = num2str(functiondg);
% need to convert the answer back into String type to display it
set(handles.answer_temeldairesicapi_staticText, 'String', dg);
guidata(hObject, handles);

functiondf = str2num(m)*str2num(z)-str2num(m)*2.4;
df = num2str(functiondf);
% need to convert the answer back into String type to display it
set(handles.answer_tabandairesicapi_staticText, 'String', df);
guidata(hObject, handles);

```

```

functions = pi*str2num(m)/2;
s = num2str(functions);
% need to convert the answer back into String type to display it
set(handles.answer_disgenisligi_staticText, 'String', s);
guidata(hObject, handles);

functionteta = 360/str2num(z);
teta = num2str(functionteta);
% need to convert the answer back into String type to display it
set(handles.answer_dislerarasiaci_staticText, 'String', teta);
guidata(hObject, handles);

functionalfat = 360/2/str2num(z);
alfat = num2str(functionalfat);
% need to convert the answer back into String type to display it
set(handles.answer_tekdisinacisi_staticText, 'String', alfat);
guidata(hObject, handles);

x=[10 ; 11 ; 12 ;13 ; 14 ; 15 ; 16 ;17 ; 18 ; 19 ; 20 ; 21 ; 22 ; 23 ;
24 ; 25 ; 26 ; 27 ;28 ; 29 ; 30 ; 31 ; 32 ; 33 ; 34 ; 35 ; 36];
y=[0.69*str2num(m) ; 0.83*str2num(m) ; 0.96*str2num(m);
1.09*str2num(m); 1.22*str2num(m); 1.34*str2num(m); 1.46*str2num(m);
1.58*str2num(m); 1.69*str2num(m); 1.79*str2num(m); 1.89*str2num(m);
1.98*str2num(m); 2.06*str2num(m); 2.15*str2num(m); 2.24*str2num(m);
2.33*str2num(m); 2.42*str2num(m); 2.50*str2num(m); 2.59*str2num(m);
2.67*str2num(m); 2.76*str2num(m); 2.85*str2num(m); 2.93*str2num(m);
3.01*str2num(m); 3.09*str2num(m); 3.16*str2num(m); 3.23*str2num(m)] ;
functionr1=interp1q(x,y, str2num(z));
r1 = num2str(functionr1);
% need to convert the answer back into String type to display it
set(handles.answer_ustyay_staticText, 'String', r1);
guidata(hObject, handles);

x=[10 ; 11 ; 12 ;13 ; 14 ; 15 ; 16 ;17 ; 18 ; 19 ; 20 ; 21 ; 22 ; 23 ;
24 ; 25 ; 26 ; 27 ;28 ; 29 ; 30 ; 31 ; 32 ; 33 ; 34 ; 35 ; 36];
y=[2.28*str2num(m) ; 2.40*str2num(m) ; 2.51*str2num(m);
2.62*str2num(m); 2.72*str2num(m); 2.82*str2num(m); 2.92*str2num(m);
3.02*str2num(m); 3.12*str2num(m); 3.22*str2num(m); 3.32*str2num(m);
3.41*str2num(m); 3.49*str2num(m); 3.57*str2num(m); 3.64*str2num(m);
3.71*str2num(m); 3.78*str2num(m); 3.85*str2num(m); 3.92*str2num(m);
3.99*str2num(m); 4.06*str2num(m); 4.13*str2num(m); 4.20*str2num(m);
4.27*str2num(m); 4.33*str2num(m); 4.39*str2num(m); 4.45*str2num(m)] ;
functionr2=interp1q(x,y, str2num(z));
r2 = num2str(functionr2);
% need to convert the answer back into String type to display it
set(handles.answer_altyay_staticText, 'String', r2);
guidata(hObject, handles);

functionVa = pi*str2num(dt)/1000*str2num(m)*b/1000/str2num(z);
Va = num2str(functionVa);
% need to convert the answer back into String type to display it
set(handles.answer_tekdis_staticText, 'String', Va);
guidata(hObject, handles);

functionVg = 2*str2num(z)*str2num(Va);
Vg = num2str(functionVg);

```

```

% need to convert the answer back into String type to display it
set(handles.answer_butundisler_staticText, 'String', Vg);
guidata(hObject, handles);

functionQ =str2num(Vg)*str2num(n);
Q = num2str(functionQ);
% need to convert the answer back into String type to display it
set(handles.answer_debi_staticText, 'String', Q);
guidata(hObject, handles);

% --- Executes on button press in anaelemancizim.
function anaelemancizim_Callback(hObject, eventdata, handles)
% hObject      handle to anaelemancizim (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
%selects axes1 as the current axes, so that
%Matlab knows where to plot the data
axes(handles.axes1)

global da
global dt
global dg
global df
global m
global t
global r1
global r2
global z

Du=str2num(da);
Do=str2num(dt);
Dt=str2num(dg);
Dd=str2num(df);
M=str2num(m);
t=str2num(t);
R1=str2num(r1);
R2=str2num(r2);
z=str2num(z);

R2=(2*R2+R1)/3;
Dem=20;
et=5;
dmil=22.8;
dgobek=dmil+3;

%axis([-200 200 -200 200])
style='-';

%H=circle(center, radius, theta1, theta2, alpha, style);

%ana govdenin disinin cizimi-----
center=[Do/2 0];
baslamaAci=270; bitisAci=359;
H=circle1(center, Du/2+et, baslamaAci, bitisAci, 0, style);

```

```

H=H'; s=size(H);
cizim(1:s(1,1),:)=H(1:s(1,1),:);

center=[Do/2 0];
baslamaAci=1; bitisAci=90;
H=circle1(center,Du/2+et,baslamaAci,bitisAci,0,style);
H=H'; s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

H=[Do/2 Du/2+et ;Dem/2+et Du/2+et; Dem/2+et Du/2+et+20; -(Dem/2+et)
Du/2+et+20; -(Dem/2+et) Du/2+et; -Do/2 Du/2+et];
s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

center=[-Do/2 0];
baslamaAci=90; bitisAci=270;
H=circle1(center,Du/2+et,baslamaAci,bitisAci,0,style);
H=H'; s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

H=[-Do/2 -(Du/2+et) ;-(Dem/2+et) -(Du/2+et); -(Dem/2+et) -
(Du/2+et+20); Dem/2+et -(Du/2+et+20); Dem/2+et -(Du/2+et); Do/2 -
(Du/2+et)];
s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

fill(cizim(:,1),cizim(:,2),'m')

clear cizim
%-----

%ana govdenin icinin cizimi-----
cizim=[Dem/2 Du/2+et+20; -(Dem/2) Du/2+et+20; -Dem/2 -(Du/2+et+20);
Dem/2 -(Du/2+et+20)];
hold on
fill(cizim(:,1),cizim(:,2),'c')
hold off

clear cizim

center=[Do/2 0];
baslamaAci=270; bitisAci=359;
H=circle1(center,Du/2,baslamaAci,bitisAci,0,style);
H=H'; s=size(H);
cizim(1:s(1,1),:)=H(1:s(1,1),:);

center=[Do/2 0];
baslamaAci=1; bitisAci=90;
H=circle1(center,Du/2,baslamaAci,bitisAci,0,style);
H=H'; s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

H=[Do/2 Du/2 ; -Do/2 Du/2];
s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

```

```

center=[-Do/2 0];
baslamaAci=90; bitisAci=270;
H=circle1(center,Du/2,baslamaAci,bitisAci,0,style);
H=H'; s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

H=[-Do/2 -(Du/2) ; Do/2 -(Du/2)];
s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

hold on
fill(cizim(:,1),cizim(:,2),'c')
hold off

clear cizim
%-----

%dislilerin cizimi-----
%yatay eksene t/4 mesafede bir dogru cizildiginde bunun Do la
cakistigi
%nokta x3 ve y3 koordinatli bu noktadan R1 ve R2 yaricapli yaylarin
merkez
%noktaları tespit edilir.
x3=sqrt((Do/2)^2-(t/4)^2); y3=-t/4;

a=Dt/2; b=Do/2; c=R1;
Th=atan2(y3,x3)*180/pi;
Thc1=acosd((a^2+b^2-c^2)/(2*a*b))+Th;
xc1=(Dt/2)*cosd(Thc1); yc1=(Dt/2)*sind(Thc1);

a=Dt/2; b=Do/2; c=R2;
Thc2=acosd((a^2+b^2-c^2)/(2*a*b))+Th;
xc2=(Dt/2)*cosd(Thc2); yc2=(Dt/2)*sind(Thc2);

a=Dd/2; b=Dt/2; c=R2;
Th2=Thc1-acosd((a^2+b^2-c^2)/(2*a*b))*180/pi;
x2=(Dd/2)*cosd(Th2); y2=(Dd/2)*sind(Th2);

a=Du/2; b=Dt/2; c=R2;
Th4=Thc2-acosd((a^2+b^2-c^2)/(2*a*b));
x4=(Du/2)*cosd(Th4); y4=(Du/2)*sind(Th4);

x5=Du/2; y5=0;
x6=x4; y6=-y4;
x7=x3; y7=-y3;
x8=x2; y8=-y2;
xc3=xc1; yc3=-yc1;
xc4=xc2; yc4=-yc2;

Th1=atan2(y8,x8)*180/pi-360/z;
x1=(Dd/2)*cosd(Th1); y1=(Dd/2)*sind(Th1);

center=[0 0];

```

```

baslamaAci=Th1+360;
bitisAci=Th2+360;
H=circle1(center,Dd/2,baslamaAci,bitisAci-1,0,style);
H=H'; s=size(H);
cizim(1:s(1,1),:)=H(1:s(1,1),:);

center=[xc2 yc2];
baslamaAci=atan2(y2-yc2,x2-xc2)*180/pi+360;
bitisAci=atan2(y4-yc2,x4-xc2)*180/pi+360;
H=circle1(center,R2,baslamaAci,bitisAci+1,0,style);
H=H'; s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

center=[xc4 yc4];
baslamaAci=atan2(y6-yc4,x6-xc4)*180/pi;
bitisAci=atan2(y8-yc4,x8-xc4)*180/pi;
H=circle1(center,R2,baslamaAci+1,bitisAci,0,style);
H=H'; s=size(H); cs=size(cizim);
cizim(cs(1,1)+1:cs(1,1)+s(1,1),:)=H(1:s(1,1),:);

dis=cizim;
s=size(dis);

for k=1:2
    if k==1
        gama=0; kayma=Do/2;
    else
        if rem(z,2)==0
            gama=360/z/2; kayma=-Do/2;
        else
            gama=0 ; kayma=-Do/2;
        end
    end

    for i=0:z-1
        alpha=i*360/z+gama;
        for j=1:s(1,1)
            cizim(s(1,1)*i+j,:)= [cosd(alpha) -sind(alpha); sind(alpha)
            cosd(alpha)]*[dis(j,1);dis(j,2)];
        end
    end
    cizim(:,1)=cizim(:,1)+kayma;
    hold on
    fill(cizim(:,1),cizim(:,2),'r')
    hold off
end
clear cizim
%-----

%millerin cizimi-----
center=[Do/2 0];
H=circle1(center,dgobek/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'y')
hold off

```

```

H=circle1(center,dmil/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'b')
hold off
center=[-Do/2 0];
H=circle1(center,dgobek/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'y')
hold off
H=circle1(center,dmil/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'b')
hold off
%-----
set(gca,'DataAspectRatio',[1 1 1], 'PlotBoxAspectRatio',[1 1 1])

guidata(hObject, handles); %updates the handles

function P=circle1(center,radius,theta1,theta2,alpha,style)
%-----
% H=CIRCLE(CENTER,RADIUS,NOP,STYLE)
% This routine draws a circle with center defined as
% a vector CENTER, radius as a scaler RADIS. NOP is
% the number of points on the circle. As to STYLE,
% use it the same way as you use the rountine PLOT.
% Since the handle of the object is returned, you
% use routine SET to get the best result.
%
% Usage Examples,
%
% circle([1,3],3,1000,':');
% circle([2,4],2,1000,'--');
%
% Zhenhai Wang <zhenhai@ieee.org>
% Version 1.00
% December, 2002
%-----
NOP=360;
if (nargin <3),
    error('Please see help for INPUT DATA.');
```

```

elseif (nargin==3)
    style='b-';
end;
THETA=linspace(0,2*pi,NOP);
RHO=ones(1,NOP)*radius;
[X,Y] = pol2cart(THETA,RHO);
X=X+center(1);
Y=Y+center(2);
X1=X(theta1:theta2);
Y1=Y(theta1:theta2);
P=[cosd(alpha) -sind(alpha); sind(alpha) cosd(alpha)]*[X1;Y1];

```

Ek.3. Radyal Santrifüj Pompa Tasarım Programı

```
function varargout = radyalsantrifuj(varargin)
% RADYALSANTRIFUJ M-file for radyalsantrifuj.fig
%     RADYALSANTRIFUJ, by itself, creates a new RADYALSANTRIFUJ or
%     raises the existing
%     singleton*.
%
%     H = RADYALSANTRIFUJ returns the handle to a new RADYALSANTRIFUJ
%     or the handle to
%     the existing singleton*.
%
%     RADYALSANTRIFUJ('CALLBACK',hObject,eventData,handles,...) calls
%     the local
%     function named CALLBACK in RADYALSANTRIFUJ.M with the given
%     input arguments.
%
%     RADYALSANTRIFUJ('Property','Value',...) creates a new
%     RADYALSANTRIFUJ or raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before radyalsantrifuj_OpeningFcn gets
%     called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to radyalsantrifuj_OpeningFcn via
%     varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help radyalsantrifuj

% Last Modified by GUIDE v2.5 13-Nov-2009 01:32:04
global Q
global Vs
global D3
global D2
global Dmil
global Dgobek
global D1
global functionD2
global beta1
global beta2
global functionz

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @radyalsantrifuj_OpeningFcn, ...
                  'gui_OutputFcn',  @radyalsantrifuj_OutputFcn, ...
```



```

        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before radyalsantrifuj is made visible.
function radyalsantrifuj_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to radyalsantrifuj (see VARARGIN)

% Choose default command line output for radyalsantrifuj
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes radyalsantrifuj wait for user response (see UIRESUME)
% uiwait(handles.cizim_pushButton);

% --- Outputs from this function are returned to the command line.
function varargout = radyalsantrifuj_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function debi_editText_Callback(hObject, eventdata, handles)
% hObject    handle to debi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of debi_editText as
text
%         str2double(get(hObject,'String')) returns contents of
debi_editText as a double

%store the contents of debi_editText as a string. if the string
%is not a number then input will be empty

```

```

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default debi_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function debi_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to debi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function basmayuksekligi_editText_Callback(hObject, eventdata,
handles)
% hObject    handle to basmayuksekligi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
basmayuksekligi_editText as text
%         str2double(get(hObject,'String')) returns contents of
basmayuksekligi_editText as a double
%store the contents of basmayuksekligi_editText as a string. if the
string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default
basmayuksekligi_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function basmayuksekligi_editText_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to basmayuksekligi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function devirsayisi_editText_Callback(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
devirsayisi_editText as text
%         str2double(get(hObject,'String')) returns contents of
devirsayisi_editText as a double
%store the contents of devirsayisi_editText as a string. if the string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default devirsayisi_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function devirsayisi_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ozgulagirlik_editText_Callback(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
ozgulagirlik_editText as text
%         str2double(get(hObject,'String')) returns contents of
ozgulagirlik_editText as a double
%store the contents of ozgulagirlik_editText as a string. if the
string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

```

```

%checks to see if input is empty. if so, default ozgulagirlik_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ozgulagirlik_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function malzemegerilme_editText_Callback(hObject, eventdata, handles)
% hObject    handle to malzemegerilme_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
malzemegerilme_editText as text
%       str2double(get(hObject,'String')) returns contents of
malzemegerilme_editText as a double
%store the contents of malzemegerilme_editText as a string. if the
string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default
malzemegerilme_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function malzemegerilme_editText_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to malzemegerilme_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))

```

```

        set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in hesap_pushbutton.
function hesap_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to hesap_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Q
global Vs
global D3
global D2
global Dmil
global Dgobek
global D1
global functionD2
global beta1
global beta2
global functionz

Q = get(handles.debi_editText, 'String');
Hm = get(handles.basmayuksekligi_editText, 'String');
n = get(handles.devirsayisi_editText, 'String');
gama = get(handles.ozgulagirlik_editText, 'String');
sigma = get(handles.malzemegecilme_editText, 'String');

% P, Sigma, R, n, z, b and gamma are variables of Strings type, and need to
be converted
% to variables of Number type before they can be added together

functionnq=(str2num(n))*sqrt(str2num(Q)/1000)/((str2num(Hm))^0.75);
nq = num2str(functionnq);
% need to convert the answer back into String type to display it
set(handles.answer_tanimsayisistaticText, 'String', nq);
guidata(hObject, handles);

ns=3.65*(functionnq);

pompaverimi=0.65;
hidrolikverim=1-0.071/((str2num(Q)/1000)^0.25);
mekanikverim=0.96;
volumetrikverim=pompaverimi/hidrolikverim/mekanikverim;
Qussu=str2num(Q)/1000/(volumetrikverim);

Pmil=(str2num(gama))*(str2num(Q)/1000)*(str2num(Hm))/75/(pompaverimi);

if Pmil<=5;
    emniyetsayisi =1.25;
elseif 5<Pmil<=25;
    emniyetsayisi =1.15;
elseif Pmil>=25;
    emniyetsayisi =1.075;
end
emniyetsayisi;

```

```

functionNem = ceil(emniyetsayisi*Pmil);
Nem = num2str(functionNem);
% need to convert the answer back into String type to display it
set(handles.answer_motorgucustaticText, 'String', Nem);
guidata(hObject, handles);

%standart mil caplari 15-20-25-30-35-40-50-60-70-80-90
milcapi = ceil((36000*Nem/(str2num(n))/(str2num(sigma)))^(1/3)*10);
if milcapi<=15;
    Dmil =15;
elseif 15<milcapi<=20;
    Dmil =20;
elseif 20<milcapi<=25;
    Dmil =25;
elseif 25<milcapi<=30;
    Dmil =30;
elseif 30<milcapi<=35;
    Dmil =35;
elseif 35<milcapi<=40;
    Dmil =40;
elseif 40<milcapi<=50;
    Dmil =50;
elseif 50<milcapi<=60;
    Dmil =60;
elseif 60<milcapi<=70;
    Dmil =70;
elseif 70<milcapi<=80;
    Dmil =80;
elseif 80<milcapi<=90;
    Dmil =90;
end
Dmil;
dmil = num2str(Dmil);
% need to convert the answer back into String type to display it
set(handles.answer_milcapistaticText, 'String', dmil);
guidata(hObject, handles);

% GRAFIKTEN K1s BULUNUR
x=[10;12;13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;28;29;30;31;32;33;34;35;36;37;38;39;40];
y=[0.11;0.1116666667;0.115;0.12;0.123;0.126;0.1285714186;0.1328571429;0.1358571429;0.1392857143;0.1431428571;0.1444285714;0.1478571429;0.1491428571;0.15;0.1533333333;0.1555555556;0.1583333333;0.1627777778;0.165;0.1666666667;0.1688888889;0.1722222222;0.1733333333;0.1761111111;0.177777778;0.1805555556;0.1844444444;0.1877777778;0.1888888889] ;
K1s=interp1q(x,y,functionnq);
guidata(hObject, handles);

Cm1=K1s*sqrt(2*9.806*(str2num(Hm)))
Com=0.9*(Cm1);
A0=(Qussu)/(Com);
Dgobek=1.4*(Dmil);
Ag=pi*((Dgobek/1000)^2)/4;
Atop=A0+Ag;

functiondo=ceil((sqrt(4*(Atop)/pi))*1000);

```

```

do = num2str(functiondo);
% need to convert the answer back into String type to display it
set(handles.answer_docapistaticText, 'String', do);
guidata(hObject, handles);

D1=ceil(0.95*(functiondo));
U1=D1/1000*(str2num(n))*pi/60;

functionbeta1=ceil(atan(Cm1/U1))+2;
beta1 = num2str(functionbeta1);
% need to convert the answer back into String type to display it
set(handles.answer_betacisistaticText, 'String', beta1);
guidata(hObject, handles);

W1=Cm1/sind(functionbeta1);
Wu1=Cm1/tand(functionbeta1);
Cu1=U1-Wu1;
C1=sqrt((Cm1)^2+(Cu1)^2);
alfa1=atan(Cm1/Cu1);

D1d=ceil((functiondo)+0.75);
D1i=2*D1-D1d;
U1d=pi*D1d/1000*(str2num(n))/60;
U1i=pi*D1i/1000*str2num(n)/60;
beta1=atan(Cm1/U1d);
beta1i=atan(Cm1/U1i);

functionb1=ceil(Qussu/pi/D1*1000/Cm1/0.6*1000);
b1 = num2str(functionb1);
% need to convert the answer back into String type to display it
set(handles.answer_kanatgirisenistaticText, 'String', b1);
guidata(hObject, handles);

% kisi tablodan 1.06 bulunur tabloyu oluşturun
x=[40;50;60;70;80;90;100;110;120;130;140;150;160;170;180;190;200;210;220;230];
y=[1.25;1.19;1.15;1.09;1.05;0.99;0.96;0.93;0.88;0.84;0.81;0.78;0.74;0.72;0.69;0.66;0.64;0.62;0.60;0.58];
kisi=interp1(x,y,ns);
guidata(hObject, handles);
U2=sqrt(2*str2num(Hm)*9.806/kisi);

functionD2=ceil(60*U2/pi/str2num(n)*1000);
D2= num2str(functionD2);
% need to convert the answer back into String type to display it
set(handles.answer_d2capistaticText, 'String', D2);
guidata(hObject, handles);

%D1/D2 DEN ÇAP KONTROLÜ YAP
beta2=27;

% GRAFİKTE K2s BULUNUR

x=[10;12;13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;28;29;30;31;32;33;34;35;36;37;38;39;40];

```

```

y=[0.08125;0.0825;0.085;0.0875;0.09;0.09216216216;0.09405405405;0.0967
5675676;0.09945945946;0.1016666667;0.105;0.1066666667;0.1086666667;0.1
116666667;0.1133333333;0.115;0.1166666667;0.1183333333;0.1208571429;0.
123;0.1242857143;0.1264285714;0.1285714286;0.1311428571;0.1332857143;0
.1362857143;0.138;0.1405714286;0.1435714286;0.1457142857] ;
K2s=interp1q(x,y,functionnq);
guidata(hObject, handles);

Cm2=K2s*sqrt(2*9.81*(str2num(Hm)));
W2=Cm2/sind(beta2);
Wu2=Cm2/tand(beta2);
Cu2=U2-Wu2;
C2=sqrt((Cm2)^2+(Cu2)^2);
alfa2=atand(Cm2/Cu2);

betam=(functionbeta1+beta2)/2;

functionz=ceil(6.5*(functionD2+D1)/(functionD2-D1)*sind(betam));
z= num2str(functionz);
% need to convert the answer back into String type to display it
set(handles.answer_kanatsayisistaticText,'String',z);
guidata(hObject, handles);

lambda2=1-(functionz*(3/sind(beta2)))/pi/functionD2;

functionb2=ceil(Qussu/pi/(functionD2/1000)/lambda2/Cm2*1000);
b2= num2str(functionb2);
% need to convert the answer back into String type to display it
set(handles.answer_kanaticikisenistaticText,'String',b2);
guidata(hObject, handles);

%Kv DEĞERİ TABLODAN BULUNUR

x=[40; 50 ;60 ;70 ;80 ;90 ;100 ;110 ;120 ;130 ;140 ;150 ;160 ;170 ;180
;190 ;200];
y=[0.47; 0.46; 0.45; 0.444; 0.41; 0.390; 0.380; 0.35875; 0.3475;
0.335; 0.33375; 0.325; 0.320; 0.315; 0.310; 0.3075; 0.305];
Kv=interp1q(x,y,ns);
guidata(hObject, handles);

Vs=Kv*sqrt(2*9.81*str2num(Hm));

%D3 grafikten bulunur

x=[40; 50 ;60 ;70 ;80 ;90 ;100 ;110 ;120 ;130 ;140 ;150 ;160 ;170 ;180
;190 ;200];
y=[0.0665;0.07;0.07625;0.0825;0.09;0.095;0.1025;0.11;0.120;0.130;0.137
5;0.155;0.164375;0.1775;0.186875;0.2;0.2125];
D3=functionD2*(1+interp1q(x,y,ns));
guidata(hObject, handles);

% --- Executes on button press in cizim_pushbutton.
function cizim_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to cizim_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```



```

% handles      structure with handles and user data (see GUIDATA)

%selects axes1 as the current axes, so that
%Matlab knows where to plot the data
axes(handles.axes1)

%axis([-200 200 -200 200])
style='-';

%H=circle(center,radius,theta1,theta2,alpha,style);

%-----
----
global Q
global Vs
global D3
global D2
global Dmil
global Dgobek
global D1
global functionD2
global beta1
global beta2
global functionz

%-----
----
%SALYANGOZ ÇİZİMİ

Q = str2num(Q)/1000;
D2= str2num(D2);

d=zeros(360,1);
x=zeros(360,1);
y=zeros(360,1);
for i=1:360
    d=sqrt(i*4*Q/360/Vs/pi)*1000;
    x(i)=(D3/2+d)*cosd(i); y(i)=(D3/2+d)*sind(i);
end
x0=0; y0=0;

%-----
----

x(361)=(D3/2+d)+(D3/2+d)*tand(10); y(361)=(D3/2+d);
x(362)=(D3/2)-(D3/2+d)*tand(10); y(362)=(D3/2+d);

cizim=[x(40:362) y(40:362)];
cizim(324,1)=x(40);
cizim(324,2)=y(40);

fill(cizim(:,1),cizim(:,2),'b')
hold off
clear cizim

```

```

%-----
% D3 ÇAPININ ÇİZİMİ

center=[x0 y0];
H=circle1(center,D3/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'m')
hold off
clear cizim
%-----

% D2 ÇAPININ ÇİZİMİ
center=[x0 y0];
H=circle1(center,D2/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'c')
hold off
clear cizim
%-----

% D1 ÇAPININ ÇİZİMİ

center=[x0 y0];
H=circle1(center,D1/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'m')
hold off
clear cizim

%-----

% KANAT ÇİZİMİ

r1=D1/20;
r2=functionD2/20;

deltateta(1)=0;
for i=2:10
    deltateta(i)=deltateta(i-1)+180/pi*((r2-r1)/10)/((r1+(i)*(r2-
r1)/10)*tand(beta1+(i)*(beta2-beta1)/10));
end
deltateta(11)=deltateta(10)+180/pi*(r2-r1)/10/r2/tand(beta2);

x(362)=10*r1-1; y(362)=0-1;
x(363)=10*r1; y(363)=0;
for i=2:10
    x(i+362)=(10*r1+(i-1)*10*(r2-r1)/10)*cosd(-deltateta(i));
y(i+362)=(10*r1+(i-1)*10*(r2-r1)/10)*sind(-deltateta(i));
end
x(373)=(10*r2*cosd(-deltateta(11))); y(373)=(10*r2*sind(-
deltateta(11)));
x(374)=x(373)+1; y(374)=y(373)+1;

cizim=[x(362:374) y(362:374)];

```

```

%Kanatları Kanat Sayısına Göre Çoğaltma

for i=0:functionz-1

    alpha=i*360/functionz;
    cizim1=cizim*[cosd(alpha) -sind(alpha); sind(alpha) cosd(alpha)];
    hold on
    plot(cizim1(:,1),cizim1(:,2),'m','linewidth',5)
    hold off

end

clear cizim

%-----

% Dgobek ÇİZİMİ
center=[x0 y0];
H=circle1(center,Dgobek/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'c')
hold off
clear cizim
%-----

% Dmil ÇİZİMİ
center=[x0 y0];
H=circle1(center,Dmil/2,1,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'y')
hold off
clear cizim

set(gca,'DataAspectRatio',[1 1 1], 'PlotBoxAspectRatio',[1 1 1])

guidata(hObject, handles); %updates the handles

function P=circle1(center,radius,theta1,theta2,alpha,style)
%-----
%-----
% H=CIRCLE(CENTER,RADIUS,NOP,STYLE)
% This routine draws a circle with center defined as
% a vector CENTER, radius as a scaler RADIS. NOP is
% the number of points on the circle. As to STYLE,
% use it the same way as you use the routine PLOT.
% Since the handle of the object is returned, you
% use routine SET to get the best result.
%
% Usage Examples,
%
% circle([1,3],3,1000,':');

```

```

%   circle([2,4],2,1000,'--');
%
%   Zhenhai Wang <zhenhai@ieee.org>
%   Version 1.00
%   December, 2002
%-----
-----
NOP=360;
if (nargin <3),
    error('Please see help for INPUT DATA. ');
elseif (nargin==3)
    style='b-';
end;
THETA=linspace(0,2*pi,NOP);
RHO=ones(1,NOP)*radius;
[X,Y] = pol2cart(THETA,RHO);
X=X+center(1);
Y=Y+center(2);
X1=X(theta1:theta2);
Y1=Y(theta1:theta2);
P=[cosd(alpha) -sind(alpha); sind(alpha) cosd(alpha)]*[X1;Y1];

```

Ek.4. Eksenel Santrifüj Pompa Tasarım Programı

```
function varargout = eksenel_santrifuj(varargin)
% EKSENEL_SANTRIFUJ M-file for eksenel_santrifuj.fig
%     EKSENEL_SANTRIFUJ, by itself, creates a new EKSENEL_SANTRIFUJ
or raises the existing
%     singleton*.
%
%     H = EKSENEL_SANTRIFUJ returns the handle to a new
EKSENEL_SANTRIFUJ or the handle to
%     the existing singleton*.
%
%     EKSENEL_SANTRIFUJ('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in EKSENEL_SANTRIFUJ.M with the given
input arguments.
%
%     EKSENEL_SANTRIFUJ('Property','Value',...) creates a new
EKSENEL_SANTRIFUJ or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before eksenel_santrifuj_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to eksenel_santrifuj_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help eksenel_santrifuj

% Last Modified by GUIDE v2.5 06-Jan-2010 15:58:33

global lA1A2
global betaalfaA1A2
global lE1E2
global betaalfaE1E2
global d2
global dh

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename, ...
                  'gui_Singleton',      gui_Singleton, ...
                  'gui_OpeningFcn',     @eksenel_santrifuj_OpeningFcn,
                  ...
                  'gui_OutputFcn',     @eksenel_santrifuj_OutputFcn, ...
                  'gui_LayoutFcn',     [] , ...
                  'gui_Callback',      []);
```

```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before eksenel_santrifuj is made visible.
function eksenel_santrifuj_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to eksenel_santrifuj (see
VARARGIN)

% Choose default command line output for eksenel_santrifuj
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes eksenel_santrifuj wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = eksenel_santrifuj_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function debi_editText_Callback(hObject, eventdata, handles)
% hObject    handle to debi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of debi_editText as
text
%         str2double(get(hObject,'String')) returns contents of
debi_editText as a double
%store the contents of debi_editText as a string. if the string
%is not a number then input will be empty

```

```

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default debi_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function debi_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to debi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function basmayuksekligi_editText_Callback(hObject, eventdata,
handles)
% hObject    handle to basmayuksekligi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
basmayuksekligi_editText as text
%         str2double(get(hObject,'String')) returns contents of
basmayuksekligi_editText as a double
%store the contents of basmayuksekligi_editText as a string. if the
string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default
basmayuksekligi_editText to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function basmayuksekligi_editText_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to basmayuksekligi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function devirsayisi_editText_Callback(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
devirsayisi_editText as text
%         str2double(get(hObject,'String')) returns contents of
devirsayisi_editText as a double

%store the contents of devirsayisi_editText as a string. if the string
%is not a number then input will be empty

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default devirsayisi_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function devirsayisi_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to devirsayisi_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ozgulagirlik_editText_Callback(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
ozgulagirlik_editText as text
%         str2double(get(hObject,'String')) returns contents of
ozgulagirlik_editText as a double
%         str2double(get(hObject,'String')) returns contents of
ozgulagirlik_editText as a double
%store the contents of ozgulagirlik_editText as a string. if the
string
%is not a number then input will be empty

```



```

input = str2num(get(hObject,'String'));

%checks to see if input is empty. if so, default ozgulagirlik_editText
to zero
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ozgulagirlik_editText_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ozgulagirlik_editText (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in hesap_pushbutton.
function hesap_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to hesap_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in hesap_pushbutton.

global lA1A2
global betaalfaA1A2
global lE1E2
global betaalfaE1E2
global d2
global dh

Q = get(handles.debi_editText,'String');
Hm = get(handles.basmayuksekligi_editText,'String');
n = get(handles.devirsayisi_editText,'String');
ozgulagirlik = get(handles.ozgulagirlik_editText,'String');
% Q,Hm,n are variables of Strings type, and need to be converted
% to variables of Number type before they can be added together

nsQ=str2num(n)*sqrt(str2num(Q)/1000)/(str2num(Hm)^0.75);
nsf=3*nsQ;

x=[600;625;650;675;700;725;750;775;800;825;850;875;900;925;950;975;100
0;1025;1050;1075;1100];
y=[0.65;0.60;0.57;0.53;0.505;0.47;0.468;0.465;0.46;0.45;0.43;0.42;0.41
;0.40;0.39;0.385;0.38;0.37;0.36;0.355;0.35];
Km=interp1q(x,y,nsf);

```

```

guidata(hObject, handles);

Cm=Km*sqrt(2*9.801*str2num(Hm));
Qussu=1.06*(str2num(Q)/1000);
Am=Qussu/Cm;
d2=sqrt(Am*4/pi/(1-0.45^2));
dh=0.45*d2;

Hth=str2num(Hm)/0.83;

%% A1A2 ARALIĞINDAKİ DEĞERLER

functiondA1A2 = dh;
dA1A2 = num2str(functiondA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_dA1A2, 'String', functiondA1A2);
guidata(hObject, handles);

functionuA1A2 = pi*functiondA1A2*str2num(n)/60;
uA1A2 = num2str(functionuA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_uA1A2, 'String', functionuA1A2);
guidata(hObject, handles);

functionCu3A1A2 = 9.806*Hth/functionuA1A2;
Cu3A1A2 = num2str(functionCu3A1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_Cu3A1A2, 'String', functionCu3A1A2);
guidata(hObject, handles);

functionw2A1A2 = Cm^2+(functionuA1A2-functionCu3A1A2/2)^2;
w2A1A2 = num2str(functionw2A1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_w2A1A2, 'String', functionw2A1A2);
guidata(hObject, handles);

functiontanbetaA1A2 = Cm/(functionuA1A2-functionCu3A1A2/2);
tanbetaA1A2 = num2str(functiontanbetaA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tanbetaA1A2, 'String', functiontanbetaA1A2);
guidata(hObject, handles);

functionbetaA1A2 = atand(functiontanbetaA1A2);
betaA1A2 = num2str(functionbetaA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaA1A2, 'String', functionbetaA1A2);
guidata(hObject, handles);

functionlamdakabulA1A2 = 1;
lamdakabulA1A2 = num2str(functionlamdakabulA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdakabulA1A2, 'String', functionlamdakabulA1A2);
guidata(hObject, handles);

```

```

functionCLltA1A2 =
2*9.806*Hth*Cm*cosd(functionlamdakabulA1A2)/(functionw2A1A2*functionuA
1A2*sind(functionbetaA1A2+functionlamdakabulA1A2));
CLltA1A2 = num2str(functionCLltA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLltA1A2, 'String', functionCLltA1A2);
guidata(hObject, handles);

functionltkabulA1A2 = 0.9;
ltkabulA1A2 = num2str(functionltkabulA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ltkabulA1A2, 'String', functionltkabulA1A2
);
guidata(hObject, handles);

functionCLA1A2 = functionCLltA1A2/functionltkabulA1A2;
CLA1A2 = num2str(functionCLA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLA1A2, 'String', functionCLA1A2);
guidata(hObject, handles);

functionProfilkabulA1A2 = 387;
ProfilkabulA1A2 = num2str(functionProfilkabulA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ProfilkabulA1A2, 'String', functionProfilk
abulA1A2);
guidata(hObject, handles);

functionymaxlkabulA1A2 = 0.1505;
ymaxlkabulA1A2 = num2str(functionymaxlkabulA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxlA1A2, 'String', functionymaxlkabulA1A
2);
guidata(hObject, handles);

functiontA1A2 = pi*functiondA1A2/3*1000;
tA1A2 = num2str(functiontA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tA1A2, 'String', functiontA1A2);
guidata(hObject, handles);

functionlA1A2 = functiontA1A2*functionltkabulA1A2;
lA1A2 = num2str(functionlA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lA1A2, 'String', functionlA1A2);
guidata(hObject, handles);

functionymaxA1A2 = functionymaxlkabulA1A2*functionlA1A2;
ymaxA1A2 = num2str(functionymaxA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxA1A2, 'String', functionymaxA1A2);
guidata(hObject, handles);

functionlamdahesapA1A2 =
atand(0.012+0.06*(functionymaxA1A2/functionlA1A2));

```

```

lamdahesapA1A2 = num2str(functionlamdahesapA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdahesapA1A2, 'String', functionlamdahesapA1A2);
guidata(hObject, handles);

functionalfaformulA1A2 = (functionCLA1A2-
4.4*functionymaxA1A2/functionlA1A2)/0.092;
alfaformulA1A2 = num2str(functionalfaformulA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_alfaformulA1A2, 'String', functionalfaformulA1A2);
guidata(hObject, handles);

functionbetaalfaA1A2 = functionbetaA1A2+functionalfaformulA1A2;
betaalfaA1A2 = num2str(functionbetaalfaA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaalfaA1A2, 'String', functionbetaalfaA1A2);
guidata(hObject, handles);

functiondeltahA1A2 = 0.7*functionCLA1A2*functionw2A1A2/9.806;
deltahA1A2 = num2str(functiondeltahA1A2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_deltahA1A2, 'String', functiondeltahA1A2);
guidata(hObject, handles);

%% B1B2 ARALIĞINDAKİ DEĞERLER

functiondB1B2 = dh+(d2-dh)/3;
dB1B2 = num2str(functiondB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_dB1B2, 'String', functiondB1B2);
guidata(hObject, handles);

functionuB1B2 = pi*functiondB1B2*str2num(n)/60;
uB1B2 = num2str(functionuB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_uB1B2, 'String', functionuB1B2);
guidata(hObject, handles);

functionCu3B1B2 = 9.806*Hth/functionuB1B2;
Cu3B1B2 = num2str(functionCu3B1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_Cu3B1B2, 'String', functionCu3B1B2);
guidata(hObject, handles);

functionw2B1B2 = Cm^2+(functionuB1B2-functionCu3B1B2/2)^2;
w2B1B2 = num2str(functionw2B1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_w2B1B2, 'String', functionw2B1B2);
guidata(hObject, handles);

functiontanbetaB1B2 = Cm/(functionuB1B2-functionCu3B1B2/2);
tanbetaB1B2 = num2str(functiontanbetaB1B2);

```

```

% need to convert the answer back into String type to display it
set(handles.answer_statictext_tanbetaB1B2, 'String', functiontanbetaB1B2
);
guidata(hObject, handles);

functionbetaB1B2 = atand(functiontanbetaB1B2);
betaB1B2 = num2str(functionbetaB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaB1B2, 'String', functionbetaB1B2);
guidata(hObject, handles);

functionlamdakabulB1B2 = 1;
lamdakabulB1B2 = num2str(functionlamdakabulB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdakabulB1B2, 'String', functionlamdakab
ulB1B2);
guidata(hObject, handles);

functionCLltB1B2 =
2*9.806*Hth*Cm*cosd(functionlamdakabulB1B2)/(functionw2B1B2*functionuB
1B2*sind(functionbetaB1B2+functionlamdakabulB1B2));
CLltB1B2 = num2str(functionCLltB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLltB1B2, 'String', functionCLltB1B2);
guidata(hObject, handles);

functionltkabulB1B2 = 0.830;
ltkabulB1B2 = num2str(functionltkabulB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ltkabulB1B2, 'String', functionltkabulB1B2
);
guidata(hObject, handles);

functionCLB1B2 = functionCLltB1B2/functionltkabulB1B2;
CLB1B2 = num2str(functionCLB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLB1B2, 'String', functionCLB1B2);
guidata(hObject, handles);

functionProfilkabulB1B2 = 490;
ProfilkabulB1B2 = num2str(functionProfilkabulB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ProfilkabulB1B2, 'String', functionProfilk
abulB1B2);
guidata(hObject, handles);

functionymaxlkabulB1B2 = 0.11;
ymaxlkabulB1B2 = num2str(functionymaxlkabulB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxlB1B2, 'String', functionymaxlkabulB1B
2);
guidata(hObject, handles);

functiontB1B2 = pi*functiondB1B2/3*1000;
tB1B2 = num2str(functiontB1B2);
% need to convert the answer back into String type to display it

```

```

set(handles.answer_statictext_tB1B2, 'String', functiontB1B2);
guidata(hObject, handles);

functionlB1B2 = functiontB1B2*functionltkabulB1B2;
lB1B2 = num2str(functionlB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lB1B2, 'String', functionlB1B2);
guidata(hObject, handles);

functionymaxB1B2 = functionymaxlkabulB1B2*functionlB1B2;
ymaxB1B2 = num2str(functionymaxB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxB1B2, 'String', functionymaxB1B2);
guidata(hObject, handles);

functionlamdahesapB1B2 =
atand(0.012+0.06*(functionymaxB1B2/functionlB1B2));
lamdahesapB1B2 = num2str(functionlamdahesapB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdahesapB1B2, 'String', functionlamdahesapB1B2);
guidata(hObject, handles);

functionalfaformulB1B2 = (functionCLB1B2-
4.4*functionymaxB1B2/functionlB1B2)/0.092;
alfaformulB1B2 = num2str(functionalfaformulB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_alfaformulB1B2, 'String', functionalfaformulB1B2);
guidata(hObject, handles);

functionbetaalfaB1B2 = functionbetaB1B2+functionalfaformulB1B2;
betaalfaB1B2 = num2str(functionbetaalfaB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaalfaB1B2, 'String', functionbetaalfaB1B2);
guidata(hObject, handles);

functiondeltahB1B2 = 0.7*functionCLB1B2*functionw2B1B2/9.806;
deltahB1B2 = num2str(functiondeltahB1B2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_deltahB1B2, 'String', functiondeltahB1B2);
guidata(hObject, handles);

%% C1C2 ARALIĞINDAKİ DEĞERLER

functiondC1C2 = dh+2*(d2-dh)/4;
dC1C2 = num2str(functiondC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_dC1C2, 'String', functiondC1C2);
guidata(hObject, handles);

functionuC1C2 = pi*functiondC1C2*str2num(n)/60;
uC1C2 = num2str(functionuC1C2);
% need to convert the answer back into String type to display it

```

```

set(handles.answer_statictext_uC1C2, 'String', functionuC1C2);
guidata(hObject, handles);

functionCu3C1C2 = 9.806*Hth/functionuC1C2;
Cu3C1C2 = num2str(functionCu3C1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_Cu3C1C2, 'String', functionCu3C1C2);
guidata(hObject, handles);

functionw2C1C2 = Cm^2+(functionuC1C2-functionCu3C1C2/2)^2;
w2C1C2 = num2str(functionw2C1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_w2C1C2, 'String', functionw2C1C2);
guidata(hObject, handles);

functiontanbetaC1C2 = Cm/(functionuC1C2-functionCu3C1C2/2);
tanbetaC1C2 = num2str(functiontanbetaC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tanbetaC1C2, 'String', functiontanbetaC1C2);
guidata(hObject, handles);

functionbetaC1C2 = atand(functiontanbetaC1C2);
betaC1C2 = num2str(functionbetaC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaC1C2, 'String', functionbetaC1C2);
guidata(hObject, handles);

functionlamdakabulC1C2 = 1;
lamdakabulC1C2 = num2str(functionlamdakabulC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdakabulC1C2, 'String', functionlamdakabulC1C2);
guidata(hObject, handles);

functionCLltC1C2 =
2*9.806*Hth*Cm*cosd(functionlamdakabulC1C2)/(functionw2C1C2*functionuC1C2*sind(functionbetaC1C2+functionlamdakabulC1C2));
CLltC1C2 = num2str(functionCLltC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLltC1C2, 'String', functionCLltC1C2);
guidata(hObject, handles);

functionltkabulC1C2 = 0.77;
ltkabulC1C2 = num2str(functionltkabulC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ltkabulC1C2, 'String', functionltkabulC1C2);
guidata(hObject, handles);

functionCLC1C2 = functionCLltC1C2/functionltkabulC1C2;
CLC1C2 = num2str(functionCLC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLC1C2, 'String', functionCLC1C2);
guidata(hObject, handles);

```

```

functionProfilkabulC1C2 = 490;
ProfilkabulC1C2 = num2str(functionProfilkabulC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ProfilkabulC1C2, 'String', functionProfilkabulC1C2);
guidata(hObject, handles);

functionymaxlkabulC1C2 = 0.085;
ymaxlkabulC1C2 = num2str(functionymaxlkabulC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxlC1C2, 'String', functionymaxlkabulC1C2);
guidata(hObject, handles);

functiontC1C2 = pi*functiondC1C2/3*1000;
tC1C2 = num2str(functiontC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tC1C2, 'String', functiontC1C2);
guidata(hObject, handles);

functionlC1C2 = functiontC1C2*functionltkabulC1C2;
lC1C2 = num2str(functionlC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lC1C2, 'String', functionlC1C2);
guidata(hObject, handles);

functionymaxC1C2 = functionymaxlkabulC1C2*functionlC1C2;
ymaxC1C2 = num2str(functionymaxC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxC1C2, 'String', functionymaxC1C2);
guidata(hObject, handles);

functionlamdahesapC1C2 =
atand(0.012+0.06*(functionymaxC1C2/functionlC1C2));
lamdahesapC1C2 = num2str(functionlamdahesapC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdahesapC1C2, 'String', functionlamdahesapC1C2);
guidata(hObject, handles);

functionalfaformulC1C2 = (functionCLC1C2-
4.4*functionymaxC1C2/functionlC1C2)/0.092;
alfaformulC1C2 = num2str(functionalfaformulC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_alfaformulC1C2, 'String', functionalfaformulC1C2);
guidata(hObject, handles);

functionbetaalfaC1C2 = functionbetaC1C2+functionalfaformulC1C2;
betaalfaC1C2 = num2str(functionbetaalfaC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaalfaC1C2, 'String', functionbetaalfaC1C2);
guidata(hObject, handles);

functiondeltahC1C2 = 0.7*functionCLC1C2*functionw2C1C2/9.806;

```



```

deltahC1C2 = num2str(functiondeltahC1C2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_deltahC1C2, 'String', functiondeltahC1C2);
guidata(hObject, handles);

%% D1D2 ARALIĞINDAKİ DEĞERLER

functiondD1D2 = dh+3*(d2-dh)/4;
dD1D2 = num2str(functiondD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_dD1D2, 'String', functiondD1D2);
guidata(hObject, handles);

functionuD1D2 = pi*functiondD1D2*str2num(n)/60;
uD1D2 = num2str(functionuD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_uD1D2, 'String', functionuD1D2);
guidata(hObject, handles);

functionCu3D1D2 = 9.806*Hth/functionuD1D2;
Cu3D1D2 = num2str(functionCu3D1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_Cu3D1D2, 'String', functionCu3D1D2);
guidata(hObject, handles);

functionw2D1D2 = Cm^2+(functionuD1D2-functionCu3D1D2/2)^2;
w2D1D2 = num2str(functionw2D1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_w2D1D2, 'String', functionw2D1D2);
guidata(hObject, handles);

functiontanbetaD1D2 = Cm/(functionuD1D2-functionCu3D1D2/2);
tanbetaD1D2 = num2str(functiontanbetaD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tanbetaD1D2, 'String', functiontanbetaD1D2);
guidata(hObject, handles);

functionbetaD1D2 = atand(functiontanbetaD1D2);
betaD1D2 = num2str(functionbetaD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaD1D2, 'String', functionbetaD1D2);
guidata(hObject, handles);

functionlamdakabulD1D2 = 1;
lamdakabulD1D2 = num2str(functionlamdakabulD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdakabulD1D2, 'String', functionlamdakabulD1D2);
guidata(hObject, handles);

functionCLltd1D2 =
2*9.806*Hth*Cm*cosd(functionlamdakabulD1D2)/(functionw2D1D2*functionuD1D2*sind(functionbetaD1D2+functionlamdakabulD1D2));
CLltd1D2 = num2str(functionCLltd1D2);
% need to convert the answer back into String type to display it

```

```

set(handles.answer_statictext_CLltD1D2, 'String', functionCLltD1D2);
guidata(hObject, handles);

functionltkabulD1D2 = 0.715;
ltkabulD1D2 = num2str(functionltkabulD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ltkabulD1D2, 'String', functionltkabulD1D2);
guidata(hObject, handles);

functionCLD1D2 = functionCLltD1D2/functionltkabulD1D2;
CLD1D2 = num2str(functionCLD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLD1D2, 'String', functionCLD1D2);
guidata(hObject, handles);

functionProfilkabulD1D2 = 490;
ProfilkabulD1D2 = num2str(functionProfilkabulD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ProfilkabulD1D2, 'String', functionProfilkabulD1D2);
guidata(hObject, handles);

functionymaxlkabulD1D2 = 0.067;
ymaxlkabulD1D2 = num2str(functionymaxlkabulD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxlkabulD1D2, 'String', functionymaxlkabulD1D2);
guidata(hObject, handles);

functiontD1D2 = pi*functiondD1D2/3*1000;
tD1D2 = num2str(functiontD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tD1D2, 'String', functiontD1D2);
guidata(hObject, handles);

functionlD1D2 = functiontD1D2*functionltkabulD1D2;
lD1D2 = num2str(functionlD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lD1D2, 'String', functionlD1D2);
guidata(hObject, handles);

functionymaxD1D2 = functionymaxlkabulD1D2*functionlD1D2;
ymaxD1D2 = num2str(functionymaxD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxD1D2, 'String', functionymaxD1D2);
guidata(hObject, handles);

functionlamdahesapD1D2 =
atand(0.012+0.06*(functionymaxD1D2/functionlD1D2));
lamdahesapD1D2 = num2str(functionlamdahesapD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdahesapD1D2, 'String', functionlamdahesapD1D2);
guidata(hObject, handles);

```

```

functionalalfaformulD1D2 = (functionCLD1D2-
4.4*functionymaxD1D2/functionlD1D2)/0.092;
alfaformulD1D2 = num2str(functionalfaformulD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_alfaformulD1D2, 'String', functionalalfaformulD1D2);
guidata(hObject, handles);

functionbetaalfaD1D2 = functionbetaD1D2+functionalalfaformulD1D2;
betaalfaD1D2 = num2str(functionbetaalfaD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaalfaD1D2, 'String', functionbetaalfaD1D2);
guidata(hObject, handles);

functiondeltahD1D2 = 0.7*functionCLD1D2*functionw2D1D2/9.806;
deltahD1D2 = num2str(functiondeltahD1D2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_deltahD1D2, 'String', functiondeltahD1D2);
guidata(hObject, handles);

%% E1E2 ARALIĞINDAKİ DEĞERLER

functiondE1E2 = d2;
dE1E2 = num2str(functiondE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_dE1E2, 'String', functiondE1E2);
guidata(hObject, handles);

functionuE1E2 = pi*functiondE1E2*str2num(n)/60;
uE1E2 = num2str(functionuE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_uE1E2, 'String', functionuE1E2);
guidata(hObject, handles);

functionCu3E1E2 = 9.806*Hth/functionuE1E2;
Cu3E1E2 = num2str(functionCu3E1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_Cu3E1E2, 'String', functionCu3E1E2);
guidata(hObject, handles);

functionw2E1E2 = Cm^2+(functionuE1E2-functionCu3E1E2/2)^2;
w2E1E2 = num2str(functionw2E1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_w2E1E2, 'String', functionw2E1E2);
guidata(hObject, handles);

functiontanbetaE1E2 = Cm/(functionuE1E2-functionCu3E1E2/2);
tanbetaE1E2 = num2str(functiontanbetaE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tanbetaE1E2, 'String', functiontanbetaE1E2);
guidata(hObject, handles);

functionbetaE1E2 = atand(functiontanbetaE1E2);
betaE1E2 = num2str(functionbetaE1E2);

```

```

% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaE1E2, 'String', functionbetaE1E2);
guidata(hObject, handles);

functionlamdakabule1E2 = 1;
lamdakabule1E2 = num2str(functionlamdakabule1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdakabule1E2, 'String', functionlamdakabule1E2);
guidata(hObject, handles);

functionCLlE1E2 =
2*9.806*Hth*Cm*cosd(functionlamdakabule1E2)/(functionw2E1E2*functionuE1E2*sind(functionbetaE1E2+functionlamdakabule1E2));
CLlE1E2 = num2str(functionCLlE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLlE1E2, 'String', functionCLlE1E2);
guidata(hObject, handles);

functionltkabule1E2 = 0.67;
ltkabule1E2 = num2str(functionltkabule1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ltkabule1E2, 'String', functionltkabule1E2);
guidata(hObject, handles);

functionCLE1E2 = functionCLlE1E2/functionltkabule1E2;
CLE1E2 = num2str(functionCLE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_CLE1E2, 'String', functionCLE1E2);
guidata(hObject, handles);

functionProfilkabule1E2 = 490;
Profilkabule1E2 = num2str(functionProfilkabule1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_Profilkabule1E2, 'String', functionProfilkabule1E2);
guidata(hObject, handles);

functionymaxlkabule1E2 = 0.06;
ymaxlkabule1E2 = num2str(functionymaxlkabule1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxlE1E2, 'String', functionymaxlkabule1E2);
guidata(hObject, handles);

functiontE1E2 = pi*functiondE1E2/3*1000;
tE1E2 = num2str(functiontE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_tE1E2, 'String', functiontE1E2);
guidata(hObject, handles);

functionlE1E2 = functiontE1E2*functionltkabule1E2;
lE1E2 = num2str(functionlE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lE1E2, 'String', functionlE1E2);

```

```

guidata(hObject, handles);

functionymaxE1E2 = functionmaxlkabuleE1E2*functionlE1E2;
ymaxE1E2 = num2str(functionymaxE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_ymaxE1E2, 'String',functionymaxE1E2);
guidata(hObject, handles);

functionlamdahesapE1E2 =
atand(0.012+0.06*(functionymaxE1E2/functionlE1E2));
lamdahesapE1E2 = num2str(functionlamdahesapE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_lamdahesapE1E2, 'String',functionlamdahes
apE1E2);
guidata(hObject, handles);

functionalfaformuleE1E2 = (functionCLE1E2-
4.4*functionymaxE1E2/functionlE1E2)/0.092;
alfaformuleE1E2 = num2str(functionalfaformuleE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_alfaformuleE1E2, 'String',functionalfaform
uleE1E2);
guidata(hObject, handles);

functionbetaalfaE1E2 = functionbetaE1E2+functionalfaformuleE1E2;
betaalfaE1E2 = num2str(functionbetaalfaE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_betaalfaE1E2, 'String',functionbetaalfaE1
E2);
guidata(hObject, handles);

functiondeltahE1E2 = 0.7*functionCLE1E2*functionw2E1E2/9.806;
deltahE1E2 = num2str(functiondeltahE1E2);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_deltahE1E2, 'String',functiondeltahE1E2);
guidata(hObject, handles);

%KANAT SAYISININ HESAPLANMASI

if 950<nsf<=1100;
    kanatsayisi =3;
elseif 800<nsf<=950;
    kanatsayisi =4;
elseif 600<nsf<=800;
    kanatsayisi =5;
end

kanatsayisi;

functionz = ceil(kanatsayisi);
z = num2str(functionz);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_zA1A2, 'String', z);
guidata(hObject, handles);

```

```

%MOTOR GÜCÜNÜN HESABI

Pmil=str2num(ozgulagirlik)*(str2num(Q)/1000)*(str2num(Hm))/75/0.95;

if Pmil<=5;
    emniyetsayisi =1.25;
elseif 5<Pmil<=25;
    emniyetsayisi =1.15;
elseif Pmil>=25;
    emniyetsayisi =1.075;
end
emniyetsayisi;

functionNem = ceil(emniyetsayisi*Pmil);
Nem = num2str(functionNem);
% need to convert the answer back into String type to display it
set(handles.answer_statictext_NmA1A2, 'String', Nem);
guidata(hObject, handles);

% --- Executes on button press in cizim_pushbutton.
function cizim_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to cizim_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%selects axes1 as the current axes, so that
%Matlab knows where to plot the data

axes(handles.axes1)

%axis([-200 200 -200 200])
style='-';

%H=circle(center, radius, theta1, theta2, alpha, style);

%-----
----
global lA1A2
global betaalfaA1A2
global lE1E2
global betaalfaE1E2
global d2
global dh

%-----
%GÖVDE ÇİZİMİ

lA1A2=str2num(lA1A2);
betaalfaA1A2=str2num(betaalfaA1A2);

lE1E2=str2num(lE1E2);
betaalfaE1E2=str2num(betaalfaE1E2);

```

```

xAlA2(1)=-0.3700*1A1A2*cosd(betaalfaA1A2)+3.20*cosd(90-betaalfaA1A2);
yAlA2(1)=-0.3700*1A1A2*sind(betaalfaA1A2)-3.20*sind(90-betaalfaA1A2);

xAlA2(17)=0.63*1A1A2*cosd(betaalfaA1A2)-0.15*cosd(90-betaalfaA1A2);
yAlA2(17)=0.63*1A1A2*sind(betaalfaA1A2)+0.15*sind(90-betaalfaA1A2);

xE1E2(1)=-0.3700*1E1E2*cosd(betaalfaE1E2)+2.00*cosd(90-betaalfaE1E2);
yE1E2(1)=-0.3700*1E1E2*sind(betaalfaE1E2)-2.00*sind(90-betaalfaE1E2);

xE1E2(17)=0.63*1E1E2*cosd(betaalfaE1E2)-0.15*cosd(90-betaalfaE1E2);
yE1E2(17)=0.63*1E1E2*sind(betaalfaE1E2)+0.15*sind(90-betaalfaE1E2);

x(1)=-d2*1000/2;      y(1)=yE1E2(1);
x(2)=-dh*1000/2;     y(2)=yAlA2(1);
x(3)=-dh*1000/2;     y(3)=yAlA2(17);
x(4)=-d2*1000/2;     y(4)=yE1E2(17);
x(5)=d2*1000/2;      y(5)=yE1E2(1);
x(6)=dh*1000/2;      y(6)=yAlA2(1);
x(7)=dh*1000/2;      y(7)=yAlA2(17);
x(8)=d2*1000/2;      y(8)=yE1E2(17);
x(9)=-dh*1000/2;     y(9)=y(2)-25;
x(10)=dh*1000/2;     y(10)=y(2)-25;
x(11)=dh*1000/2;     y(11)=y(7)+20;
x(12)=-dh*1000/2;    y(12)=y(3)+20;
x(13)=0;              y(13)=y(2)-25;

cizim=[x(9) y(9);
        x(10) y(10);
        x(11) y(11);
        x(12) y(12)];

fill(cizim(:,1),cizim(:,2),'c')
hold on
hold off
clear cizim

center=[x(13) y(13)];
H=circle1(center,dh*1000/2,180,360,0,style);
cizim=H';
hold on
fill(cizim(:,1),cizim(:,2),'c')
hold off
clear cizim
%-----

%KANATLARIN YANDAN GÖRÜNÜŞÜ
cizim=[x(1) y(1);
        x(2) y(2);
        x(3) y(3);
        x(4) y(4);
        x(1) y(1)];

```

```
hold on
fill(cizim(:,1),cizim(:,2),[1,0.4,0.3])
hold off
clear cizim
```

```
cizim=[x(5) y(5);
       x(6) y(6);
       x(7) y(7);
       x(8) y(8);
       x(5) y(5)];
```

```
hold on
fill(cizim(:,1),cizim(:,2),[1,0.4,0.3])
hold off
```

```
clear cizim
```

```
%-----
```

```
%KANAT ÇİZİMİ
```

```
cizim=[xA1A2(1) yA1A2(1);
       xA1A2(17) yA1A2(17);
       xE1E2(17) yE1E2(17);
       xE1E2(1) yE1E2(1)];
```

```
hold on
fill(cizim(:,1),cizim(:,2),'r')
hold off
clear cizim
```

```
%-----
```

```
%A1A2 KANAT KESİTİNİN ÇİZİLMESİ
```

```
xA1A2(1)=-0.3700*lA1A2*cosd(betaalfaA1A2)+3.20*cosd(90-betaalfaA1A2);
yA1A2(1)=-0.3700*lA1A2*sind(betaalfaA1A2)-3.20*sind(90-betaalfaA1A2);
```

```
xA1A2(2)=-0.3575*lA1A2*cosd(betaalfaA1A2)+1.50*cosd(90-betaalfaA1A2);
yA1A2(2)=-0.3575*lA1A2*sind(betaalfaA1A2)-1.50*sind(90-betaalfaA1A2);
```

```
xA1A2(3)=-0.3450*lA1A2*cosd(betaalfaA1A2)+1.05*cosd(90-betaalfaA1A2);
yA1A2(3)=-0.3450*lA1A2*sind(betaalfaA1A2)-1.05*sind(90-betaalfaA1A2);
```

```
xA1A2(4)=-0.3200*lA1A2*cosd(betaalfaA1A2)+0.55*cosd(90-betaalfaA1A2);
yA1A2(4)=-0.3200*lA1A2*sind(betaalfaA1A2)-0.55*sind(90-betaalfaA1A2);
```

```
xA1A2(5)=-0.2950*lA1A2*cosd(betaalfaA1A2)+0.25*cosd(90-betaalfaA1A2);
yA1A2(5)=-0.2950*lA1A2*sind(betaalfaA1A2)-0.25*sind(90-betaalfaA1A2);
```

```
xA1A2(6)=-0.2700*lA1A2*cosd(betaalfaA1A2)+0.10*cosd(90-betaalfaA1A2);
yA1A2(6)=-0.2700*lA1A2*sind(betaalfaA1A2)-0.10*sind(90-betaalfaA1A2);
```

```
xA1A2(7)=-0.2200*lA1A2*cosd(betaalfaA1A2)+0.00*cosd(90-betaalfaA1A2);
yA1A2(7)=-0.2200*lA1A2*sind(betaalfaA1A2)-0.00*sind(90-betaalfaA1A2);
```

```
xA1A2(8)=-0.1700*lA1A2*cosd(betaalfaA1A2)+0.00*cosd(90-betaalfaA1A2);
yA1A2(8)=-0.1700*lA1A2*sind(betaalfaA1A2)-0.00*sind(90-betaalfaA1A2);
```



```

xA1A2 (9)=-0.0070*1A1A2*cosd(betaalfaA1A2)+0.00*cosd(90-betaalfaA1A2);
yA1A2 (9)=-0.0070*1A1A2*sind(betaalfaA1A2)-0.00*sind(90-betaalfaA1A2);

xA1A2 (10)=0.03*1A1A2*cosd(betaalfaA1A2)-0.40*cosd(90-betaalfaA1A2);
yA1A2 (10)=0.03*1A1A2*sind(betaalfaA1A2)+0.40*sind(90-betaalfaA1A2);

xA1A2 (11)=0.13*1A1A2*cosd(betaalfaA1A2)-0.45*cosd(90-betaalfaA1A2);
yA1A2 (11)=0.13*1A1A2*sind(betaalfaA1A2)+0.45*sind(90-betaalfaA1A2);

xA1A2 (12)=0.23*1A1A2*cosd(betaalfaA1A2)-0.50*cosd(90-betaalfaA1A2);
yA1A2 (12)=0.23*1A1A2*sind(betaalfaA1A2)+0.50*sind(90-betaalfaA1A2);

xA1A2 (13)=0.33*1A1A2*cosd(betaalfaA1A2)-0.45*cosd(90-betaalfaA1A2);
yA1A2 (13)=0.33*1A1A2*sind(betaalfaA1A2)+0.45*sind(90-betaalfaA1A2);

xA1A2 (14)=0.43*1A1A2*cosd(betaalfaA1A2)-0.30*cosd(90-betaalfaA1A2);
yA1A2 (14)=0.43*1A1A2*sind(betaalfaA1A2)+0.30*sind(90-betaalfaA1A2);

xA1A2 (15)=0.53*1A1A2*cosd(betaalfaA1A2)-0.15*cosd(90-betaalfaA1A2);
yA1A2 (15)=0.53*1A1A2*sind(betaalfaA1A2)+0.15*sind(90-betaalfaA1A2);

xA1A2 (16)=0.58*1A1A2*cosd(betaalfaA1A2)-0.05*cosd(90-betaalfaA1A2);
yA1A2 (16)=0.58*1A1A2*sind(betaalfaA1A2)+0.05*sind(90-betaalfaA1A2);

xA1A2 (17)=0.63*1A1A2*cosd(betaalfaA1A2)-0.15*cosd(90-betaalfaA1A2);
yA1A2 (17)=0.63*1A1A2*sind(betaalfaA1A2)+0.15*sind(90-betaalfaA1A2);

xA1A2 (18)=0.58*1A1A2*cosd(betaalfaA1A2)+1.75*cosd(90-betaalfaA1A2);
yA1A2 (18)=0.58*1A1A2*sind(betaalfaA1A2)-1.75*sind(90-betaalfaA1A2);

xA1A2 (19)=0.53*1A1A2*cosd(betaalfaA1A2)+3.25*cosd(90-betaalfaA1A2);
yA1A2 (19)=0.53*1A1A2*sind(betaalfaA1A2)-3.25*sind(90-betaalfaA1A2);

xA1A2 (20)=0.43*1A1A2*cosd(betaalfaA1A2)+6.15*cosd(90-betaalfaA1A2);
yA1A2 (20)=0.43*1A1A2*sind(betaalfaA1A2)-6.15*sind(90-betaalfaA1A2);

xA1A2 (21)=0.33*1A1A2*cosd(betaalfaA1A2)+8.90*cosd(90-betaalfaA1A2);
yA1A2 (21)=0.33*1A1A2*sind(betaalfaA1A2)-8.90*sind(90-betaalfaA1A2);

xA1A2 (22)=0.23*1A1A2*cosd(betaalfaA1A2)+11.35*cosd(90-betaalfaA1A2);
yA1A2 (22)=0.23*1A1A2*sind(betaalfaA1A2)-11.35*sind(90-betaalfaA1A2);

xA1A2 (23)=0.13*1A1A2*cosd(betaalfaA1A2)+13.35*cosd(90-betaalfaA1A2);
yA1A2 (23)=0.13*1A1A2*sind(betaalfaA1A2)-13.35*sind(90-betaalfaA1A2);

xA1A2 (24)=0.03*1A1A2*cosd(betaalfaA1A2)+14.60*cosd(90-betaalfaA1A2);
yA1A2 (24)=0.03*1A1A2*sind(betaalfaA1A2)-14.60*sind(90-betaalfaA1A2);

xA1A2 (25)=-0.0700*1A1A2*cosd(betaalfaA1A2)+15.05*cosd(90-
betaalfaA1A2);

```

yA1A2(25)=-0.0700*1A1A2*sind(betaalfaA1A2)-15.05*sind(90-betaalfaA1A2);

xA1A2(26)=-0.1700*1A1A2*cosd(betaalfaA1A2)+14.40*cosd(90-betaalfaA1A2);

yA1A2(26)=-0.1700*1A1A2*sind(betaalfaA1A2)-14.40*sind(90-betaalfaA1A2);

xA1A2(27)=-0.2200*1A1A2*cosd(betaalfaA1A2)+13.40*cosd(90-betaalfaA1A2);

yA1A2(27)=-0.2200*1A1A2*sind(betaalfaA1A2)-13.40*sind(90-betaalfaA1A2);

xA1A2(28)=-0.2700*1A1A2*cosd(betaalfaA1A2)+11.95*cosd(90-betaalfaA1A2);

yA1A2(28)=-0.2700*1A1A2*sind(betaalfaA1A2)-11.95*sind(90-betaalfaA1A2);

xA1A2(29)=-0.2950*1A1A2*cosd(betaalfaA1A2)+10.85*cosd(90-betaalfaA1A2);

yA1A2(29)=-0.2950*1A1A2*sind(betaalfaA1A2)-10.85*sind(90-betaalfaA1A2);

xA1A2(30)=-0.3200*1A1A2*cosd(betaalfaA1A2)+9.40*cosd(90-betaalfaA1A2);
yA1A2(30)=-0.3200*1A1A2*sind(betaalfaA1A2)-9.40*sind(90-betaalfaA1A2);

xA1A2(31)=-0.3450*1A1A2*cosd(betaalfaA1A2)+7.65*cosd(90-betaalfaA1A2);
yA1A2(31)=-0.3450*1A1A2*sind(betaalfaA1A2)-7.65*sind(90-betaalfaA1A2);

xA1A2(32)=-0.3575*1A1A2*cosd(betaalfaA1A2)+6.25*cosd(90-betaalfaA1A2);
yA1A2(32)=-0.3575*1A1A2*sind(betaalfaA1A2)-6.25*sind(90-betaalfaA1A2);

cizim=[xA1A2(1) yA1A2(1);
xA1A2(2) yA1A2(2);
xA1A2(3) yA1A2(3);
xA1A2(4) yA1A2(4);
xA1A2(5) yA1A2(5);
xA1A2(6) yA1A2(6);
xA1A2(7) yA1A2(7);
xA1A2(8) yA1A2(8);
xA1A2(9) yA1A2(9);
xA1A2(10) yA1A2(10);
xA1A2(11) yA1A2(11);
xA1A2(12) yA1A2(12);
xA1A2(13) yA1A2(13);
xA1A2(14) yA1A2(14);
xA1A2(15) yA1A2(15);
xA1A2(16) yA1A2(16);
xA1A2(17) yA1A2(17);
xA1A2(18) yA1A2(18);
xA1A2(19) yA1A2(19);
xA1A2(20) yA1A2(20);
xA1A2(21) yA1A2(21);
xA1A2(22) yA1A2(22);

```

xA1A2(23) yA1A2(23);
xA1A2(24) yA1A2(24);
xA1A2(25) yA1A2(25);
xA1A2(26) yA1A2(26);
xA1A2(27) yA1A2(27);
xA1A2(28) yA1A2(28);
xA1A2(29) yA1A2(29);
xA1A2(30) yA1A2(30);
xA1A2(31) yA1A2(31);
xA1A2(32) yA1A2(32);
xA1A2(1) yA1A2(1)];

hold on
fill(cizim(:,1),cizim(:,2),'r')
hold off

hold on
plot(cizim(:,1),cizim(:,2),'r')
hold off

clear cizim
%-----

%E1E2 KANAT KESİTİNİN ÇİZİLMESİ

xE1E2(1)=-0.3700*1E1E2*cosd(betaalfaE1E2)+2.00*cosd(90-betaalfaE1E2);
yE1E2(1)=-0.3700*1E1E2*sind(betaalfaE1E2)-2.00*sind(90-betaalfaE1E2);

xE1E2(2)=-0.3575*1E1E2*cosd(betaalfaE1E2)+0.85*cosd(90-betaalfaE1E2);
yE1E2(2)=-0.3575*1E1E2*sind(betaalfaE1E2)-0.85*sind(90-betaalfaE1E2);

xE1E2(3)=-0.3450*1E1E2*cosd(betaalfaE1E2)+0.50*cosd(90-betaalfaE1E2);
yE1E2(3)=-0.3450*1E1E2*sind(betaalfaE1E2)-0.50*sind(90-betaalfaE1E2);

xE1E2(4)=-0.3200*1E1E2*cosd(betaalfaE1E2)+0.15*cosd(90-betaalfaE1E2);
yE1E2(4)=-0.3200*1E1E2*sind(betaalfaE1E2)-0.15*sind(90-betaalfaE1E2);

xE1E2(5)=-0.2950*1E1E2*cosd(betaalfaE1E2)+0.00*cosd(90-betaalfaE1E2);
yE1E2(5)=-0.2950*1E1E2*sind(betaalfaE1E2)-0.00*sind(90-betaalfaE1E2);

xE1E2(6)=-0.2700*1E1E2*cosd(betaalfaE1E2)+0.00*cosd(90-betaalfaE1E2);
yE1E2(6)=-0.2700*1E1E2*sind(betaalfaE1E2)-0.00*sind(90-betaalfaE1E2);

xE1E2(7)=-0.2200*1E1E2*cosd(betaalfaE1E2)+0.20*cosd(90-betaalfaE1E2);
yE1E2(7)=-0.2200*1E1E2*sind(betaalfaE1E2)-0.20*sind(90-betaalfaE1E2);

xE1E2(8)=-0.1700*1E1E2*cosd(betaalfaE1E2)+0.40*cosd(90-betaalfaE1E2);
yE1E2(8)=-0.1700*1E1E2*sind(betaalfaE1E2)-0.40*sind(90-betaalfaE1E2);

xE1E2(9)=-0.0070*1E1E2*cosd(betaalfaE1E2)+0.95*cosd(90-betaalfaE1E2);
yE1E2(9)=-0.0070*1E1E2*sind(betaalfaE1E2)-0.95*sind(90-betaalfaE1E2);

xE1E2(10)=0.03*1E1E2*cosd(betaalfaE1E2)-0.80*cosd(90-betaalfaE1E2);
yE1E2(10)=0.03*1E1E2*sind(betaalfaE1E2)+0.80*sind(90-betaalfaE1E2);

```

```

xE1E2 (11)=0.13*1E1E2*cosd(betaalfaE1E2)-0.80*cosd(90-betaalfaE1E2);
yE1E2 (11)=0.13*1E1E2*sind(betaalfaE1E2)+0.80*sind(90-betaalfaE1E2);

xE1E2 (12)=0.23*1E1E2*cosd(betaalfaE1E2)-0.60*cosd(90-betaalfaE1E2);
yE1E2 (12)=0.23*1E1E2*sind(betaalfaE1E2)+0.60*sind(90-betaalfaE1E2);

xE1E2 (13)=0.33*1E1E2*cosd(betaalfaE1E2)-0.40*cosd(90-betaalfaE1E2);
yE1E2 (13)=0.33*1E1E2*sind(betaalfaE1E2)+0.40*sind(90-betaalfaE1E2);

xE1E2 (14)=0.43*1E1E2*cosd(betaalfaE1E2)-0.15*cosd(90-betaalfaE1E2);
yE1E2 (14)=0.43*1E1E2*sind(betaalfaE1E2)+0.15*sind(90-betaalfaE1E2);

xE1E2 (15)=0.53*1E1E2*cosd(betaalfaE1E2)-0.00*cosd(90-betaalfaE1E2);
yE1E2 (15)=0.53*1E1E2*sind(betaalfaE1E2)+0.00*sind(90-betaalfaE1E2);

xE1E2 (16)=0.58*1E1E2*cosd(betaalfaE1E2)-0.05*cosd(90-betaalfaE1E2);
yE1E2 (16)=0.58*1E1E2*sind(betaalfaE1E2)+0.05*sind(90-betaalfaE1E2);

xE1E2 (17)=0.63*1E1E2*cosd(betaalfaE1E2)-0.15*cosd(90-betaalfaE1E2);
yE1E2 (17)=0.63*1E1E2*sind(betaalfaE1E2)+0.15*sind(90-betaalfaE1E2);

xE1E2 (18)=0.58*1E1E2*cosd(betaalfaE1E2)+1.45*cosd(90-betaalfaE1E2);
yE1E2 (18)=0.58*1E1E2*sind(betaalfaE1E2)-1.45*sind(90-betaalfaE1E2);

xE1E2 (19)=0.53*1E1E2*cosd(betaalfaE1E2)+2.50*cosd(90-betaalfaE1E2);
yE1E2 (19)=0.53*1E1E2*sind(betaalfaE1E2)-2.50*sind(90-betaalfaE1E2);

xE1E2 (20)=0.43*1E1E2*cosd(betaalfaE1E2)+4.40*cosd(90-betaalfaE1E2);
yE1E2 (20)=0.43*1E1E2*sind(betaalfaE1E2)-4.40*sind(90-betaalfaE1E2);

xE1E2 (21)=0.33*1E1E2*cosd(betaalfaE1E2)+6.05*cosd(90-betaalfaE1E2);
yE1E2 (21)=0.33*1E1E2*sind(betaalfaE1E2)-6.05*sind(90-betaalfaE1E2);

xE1E2 (22)=0.23*1E1E2*cosd(betaalfaE1E2)+7.45*cosd(90-betaalfaE1E2);
yE1E2 (22)=0.23*1E1E2*sind(betaalfaE1E2)-7.45*sind(90-betaalfaE1E2);

xE1E2 (23)=0.13*1E1E2*cosd(betaalfaE1E2)+8.55*cosd(90-betaalfaE1E2);
yE1E2 (23)=0.13*1E1E2*sind(betaalfaE1E2)-8.55*sind(90-betaalfaE1E2);

xE1E2 (24)=0.03*1E1E2*cosd(betaalfaE1E2)+9.05*cosd(90-betaalfaE1E2);
yE1E2 (24)=0.03*1E1E2*sind(betaalfaE1E2)-9.05*sind(90-betaalfaE1E2);

xE1E2 (25)=-0.0700*1E1E2*cosd(betaalfaE1E2)+9.60*cosd(90-betaalfaE1E2);
yE1E2 (25)=-0.0700*1E1E2*sind(betaalfaE1E2)-9.60*sind(90-betaalfaE1E2);

xE1E2 (26)=-0.1700*1E1E2*cosd(betaalfaE1E2)+9.20*cosd(90-betaalfaE1E2);
yE1E2 (26)=-0.1700*1E1E2*sind(betaalfaE1E2)-9.20*sind(90-betaalfaE1E2);

xE1E2 (27)=-0.2200*1E1E2*cosd(betaalfaE1E2)+8.65*cosd(90-betaalfaE1E2);
yE1E2 (27)=-0.2200*1E1E2*sind(betaalfaE1E2)-8.65*sind(90-betaalfaE1E2);

```

```
xE1E2(28)=-0.2700*1E1E2*cosd(betaalfaE1E2)+7.70*cosd(90-betaalfaE1E2);
yE1E2(28)=-0.2700*1E1E2*sind(betaalfaE1E2)-7.70*sind(90-betaalfaE1E2);
```

```
xE1E2(29)=-0.2950*1E1E2*cosd(betaalfaE1E2)+7.00*cosd(90-betaalfaE1E2);
yE1E2(29)=-0.2950*1E1E2*sind(betaalfaE1E2)-7.00*sind(90-betaalfaE1E2);
```

```
xE1E2(30)=-0.3200*1E1E2*cosd(betaalfaE1E2)+5.95*cosd(90-betaalfaE1E2);
yE1E2(30)=-0.3200*1E1E2*sind(betaalfaE1E2)-5.95*sind(90-betaalfaE1E2);
```

```
xE1E2(31)=-0.3450*1E1E2*cosd(betaalfaE1E2)+4.60*cosd(90-betaalfaE1E2);
yE1E2(31)=-0.3450*1E1E2*sind(betaalfaE1E2)-4.60*sind(90-betaalfaE1E2);
```

```
xE1E2(32)=-0.3575*1E1E2*cosd(betaalfaE1E2)+3.60*cosd(90-betaalfaE1E2);
yE1E2(32)=-0.3575*1E1E2*sind(betaalfaE1E2)-3.60*sind(90-betaalfaE1E2);
```

```
cizim=[xE1E2(1) yE1E2(1);
       xE1E2(2) yE1E2(2);
       xE1E2(3) yE1E2(3);
       xE1E2(4) yE1E2(4);
       xE1E2(5) yE1E2(5);
       xE1E2(6) yE1E2(6);
       xE1E2(7) yE1E2(7);
       xE1E2(8) yE1E2(8);
       xE1E2(9) yE1E2(9);
       xE1E2(10) yE1E2(10);
       xE1E2(11) yE1E2(11);
       xE1E2(12) yE1E2(12);
       xE1E2(13) yE1E2(13);
       xE1E2(14) yE1E2(14);
       xE1E2(15) yE1E2(15);
       xE1E2(16) yE1E2(16);
       xE1E2(17) yE1E2(17);
       xE1E2(18) yE1E2(18);
       xE1E2(19) yE1E2(19);
       xE1E2(20) yE1E2(20);
       xE1E2(21) yE1E2(21);
       xE1E2(22) yE1E2(22);
       xE1E2(23) yE1E2(23);
       xE1E2(24) yE1E2(24);
       xE1E2(25) yE1E2(25);
       xE1E2(26) yE1E2(26);
       xE1E2(27) yE1E2(27);
       xE1E2(28) yE1E2(28);
       xE1E2(29) yE1E2(29);
       xE1E2(30) yE1E2(30);
       xE1E2(31) yE1E2(31);
       xE1E2(32) yE1E2(32);
       xE1E2(1) yE1E2(1)];
```

```
hold on
fill(cizim(:,1),cizim(:,2),[1,0.4,0.8])
hold off
clear cizim
```

```

set(gca, 'DataAspectRatio', [1 1 1], 'PlotBoxAspectRatio', [1 1 1])

guidata(hObject, handles); %updates the handles

function P=circle1(center, radius, theta1, theta2, alpha, style)
%-----
%-----
% H=CIRCLE(CENTER, RADIUS, NOP, STYLE)
% This routine draws a circle with center defined as
% a vector CENTER, radius as a scalar RADIS. NOP is
% the number of points on the circle. As to STYLE,
% use it the same way as you use the routine PLOT.
% Since the handle of the object is returned, you
% use routine SET to get the best result.
%
% Usage Examples,
%
% circle([1,3],3,1000,':');
% circle([2,4],2,1000,'--');
%
% Zhenhai Wang <zhenhai@ieee.org>
% Version 1.00
% December, 2002
%-----
%-----

NOP=360;
if (nargin <3),
    error('Please see help for INPUT DATA. ');
elseif (nargin==3)
    style='b-';
end;
THETA=linspace(0,2*pi,NOP);
RHO=ones(1,NOP)*radius;
[X,Y] = pol2cart(THETA,RHO);
X=X+center(1);
Y=Y+center(2);
X1=X(theta1:theta2);
Y1=Y(theta1:theta2);
P=[cosd(alpha) -sind(alpha); sind(alpha) cosd(alpha)]*[X1;Y1];

% --- Executes on button press in hesap_pushbutton.
function hesapdeneme_Callback(hObject, eventdata, handles)
% hObject    handle to hesap_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```