

**T.C.**  
**BALIKESİR ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**



**STL FORMATLI CAD BİLGİLERİNDEN**  
**NESNE OLUŞTURMAK**

**MEHMET İNCE**

**YÜKSEK LİSANS TEZİ**

**Jüri Üyeleri :** **Dr. Öğr. Üyesi Davut AKDAŞ** (Tez Danışmanı)  
**Doç. Dr. Sabri BIÇAKÇI**  
**Dr. Öğr. Üyesi Cemil KÖZKURT**

**BALIKESİR, OCAK- 2023**

## **ETİK BEYAN**

Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak tarafımda hazırlanan “**Stl Formatlı Cad Bilgilerinden Nesne Oluşturmak**” başlıklı tezde;

- Tüm bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Kullanılan veriler ve sonuçlarda herhangi bir değişiklik yapmadığımı,
- Tüm bilgi ve sonuçları bilimsel araştırma ve etik ilkelere uygun şekilde sunduğumu,
- Yararlandığım eserlere atıfta bulunarak kaynak gösterdiğimi,

beyan eder, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ederim.

**Mehmet İNCE**

## ÖZET

**STL FORMATLI CAD BİLGİLERİNDEN NESNE OLUŞTURMAK**  
**YÜKSEK LİSANS TEZİ**  
**MEHMET İNCE**  
**BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**  
**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**  
**(TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ DAVUT AKDAŞ)**  
**BALIKESİR, OCAK - 2023**

Günümüzde neredeyse her alanda yedek parça temininde problemler yaşanmaktadır. Bu problemlerin nedeni, üretimin talebe yetişememesi ile de olabilir veya parça gereksinimi olan ürünün imalatının artık yapılmıyor olması da olabilir. Üretimi olmayan ancak ihtiyaç duyulan mekanik parçaların veya hasar görmüş mekanik parçaların tersine mühendislik yöntemi ile modelleme oluşturularak nesnenin imal edilmesine ihtiyaç duyulmaktadır. Üretimde yapılacak bir ürünün bütün parçalarının çizim dosyalarının saklanması ve yıllar sonrasında dahi olsa ihtiyaç duyan kişinin bu dosyalar arasından parçanın çizimini alarak 3D yazıcılar aracılığı ile ilgili parçayı oluşturulması bir çözüm olabilir. Ancak çizim dosyalarının büyüklüğü düşünüldüğünde yüzlerce parçası olan ürünler için bunu olanaksız hale getirebilir. Eğer ki saklanacak dosyalar, STL formatlı bir dosya olarak düşünülürse, çizim formatlarına göre daha az yer kaplayacaktır.

Çalışmamızda tersine mühendislik yöntemi kullanarak STL formatlı dosyalardan MATLAB yazılım ortamını kullanarak nesnenin çizdirilmesi sağlanmıştır. MATLAB da çizim oluşturulduktan sonra, nesnenin herhangi bir yerinde kullanıcı istediği şekilde değişiklik yapılabilir, aynı zamanda çizim Api arayüzü kullanılarak Solidworks programında da çizdirilip kullanıma sunulmuştur. Bu konularda ülkemizde yapılacak çalışmalara temel fikir sunmak amaçlanmıştır.

**ANAHTAR KELİMELER:** Tersine mühendislik, stl format, matlab, solidwork api arayüzü, nesne oluşturmak

Bilim Kod / Kodları: 90527, 90535, 91406

Sayfa Sayısı : 34

## **ABSTRACT**

### **CREATING OBJECT FROM CAD INFORMATION WITH STL FORMAT**

**MSC THESIS**

**MEHMET İNCE**

**BALIKESİR UNIVERSITY INSTITUTE OF SCIENCE**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**(SUPERVISOR: ASSIST. PROF. DR. DAVUT AKDAŞ)**

**BALIKESİR, OCAK - 2023**

Nowadays, there are problems in the supply of spare parts in almost every field. The reason for these problems may be that the production cannot keep up with the demand, or that the product that needs parts is no longer produced. There is a need to manufacture the object by creating modeling by reverse engineering method of mechanical parts that are not produced but needed or damaged mechanical parts. A solution can be storing the drawing files of all parts of a product to be reproduced and to create the needed part through 3D printers by taking the drawing of the part from these files, even after years, who needs it. However, considering the size of the drawing files, it may make this impossible for products with hundreds of parts. If the files to be stored are considered as STL formatted files, they will take up less space than drawing formats.

In our study, the object was drawn in the MATLAB software environment from STL formatted files using reverse engineering method. After the drawing is created in MATLAB, the user can make changes in any part of the object as he wishes, and the drawing is also drawn and made available in the Solidworks program using the Api interface. It is our aim to present a basic idea for the further studies to be carried out in our country on subjects presented in this thesis.

**KEYWORDS:** Reverse engineering, stl format, matlab, solidwork api interface, creating object

Science Code / Codes: 90527, 90535, 91406

Page Number: 34

# İÇİNDEKİLER

## Sayfa

<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>İÇİNDEKİLER</b> .....	<b>iii</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>iv</b>
<b>ÖNSÖZ</b> .....	<b>v</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. TERSİNE MÜHENDİSLİK METODU</b> .....	<b>3</b>
<b>3. STL FORMATLI DOSYALAR</b> .....	<b>7</b>
<b>4. STL FORMATLI DOSYALARIN MATLABDA UYGULANMASI</b> .....	<b>11</b>
4.1 Matlab Uygulamaları .....	11
<b>5. SOLIDWORKS API ARAYÜZ UYGULAMASI</b> .....	<b>21</b>
5.1 Solidworks Api/Makro Uygulamaları .....	21
<b>6. DEĞERLENDİRME, SONUÇ ve GELECEK ÇALIŞMALALAR</b> .....	<b>27</b>
6.1 Değerlendirme ve Sonuç .....	27
6.2 Gelecek Çalışmalar .....	28
<b>7. KAYNAKLAR</b> .....	<b>29</b>
<b>EKLER</b> .....	<b>31</b>
EK A: Matlab Kodları .....	32
EK B: Solidworks Api/Makro Kodları .....	33
<b>ÖZGEÇMİŞ</b> .....	<b>34</b>

## ŞEKİL LİSTESİ

### Sayfa

Şekil 2.1: Tersine mühendislik uygulama aşamaları .....	4
Şekil 3.1: ASCII kodlama .....	7
Şekil 3.2: Binary kodlama.....	8
Şekil 3.3: İçe dönük normal hatası.....	9
Şekil 3.4: Bozuk kenarlar.....	9
Şekil 3.5: Çakışma .....	9
Şekil 3.6: Boşluklar.....	10
Şekil 4.1: Matlab içerisinde stl dosyasının açılması, yüklenmesi.....	11
Şekil 4.2: Matlab içerisinde açılmış olan stl dosyası .....	12
Şekil 4.3: Yeni bir script oluşturma .....	14
Şekil 4.4: Matlab’da fopen komutu ile stl dosyasının açılması .....	15
Şekil 4.5: “fscanf” komutu kullanılarak text içerisinde ki istenilen değerlerin alınması....	15
Şekil 4.6: 3x3 lük vertex matrilerinin oluşturulması .....	16
Şekil 4.7: Matris değerleri arasında çizgiler çizilerek üçgenlerin oluşturulması .....	17
Şekil 4.8: Programımız çalıştırılarak çizdirilmiş altıgen prizma .....	18
Şekil 4.9: Programımız çalıştırılarak çizdirilmiş küp .....	19
Şekil 4.10: Programımız çalıştırılarak çizdirilmiş daire prizma .....	19
Şekil 4.11: Programımız çalıştırılarak çizdirilmiş silindir .....	20
Şekil 4.12: Programımız çalıştırılarak çizdirilmiş altıgen delikli dikdörtgenler prizması...20	20
Şekil 5.1: Solidworks makro komutları için hiyerarşik akış sırası.....	22
Şekil 5.2: Solidworks programında yeni makro düzenleyicisinin oluşturulması .....	22
Şekil 5.3: Makro düzenleyicisi içerisinde solidworks bağlantısı tanımlanması .....	23
Şekil 5.4: Çizimin yapılması istenilen düzlemin oluşturulması.....	24
Şekil 5.5: Sketch açılması .....	24
Şekil 5.6: Kare çizimi ve katılaştırma .....	25
Şekil 5.7: Api arayüzü ile makro komutu çalıştırılarak çizdirilmiş bir küp.....	26
Şekil 5.8: Api arayüzü ile makro komutu çalıştırılarak çizdirilmiş bir silindir .....	26

## **ÖNSÖZ**

Bu tez tersine mühendislik metodu ile MATLAB program kullanılarak, ASCII kodlamaya sahip STL formatlı dosyasından parça şekillerinin oluşturulmasını içermektedir. Ülkemizde bilgisayar destekli tasarım alanlarından, tersine mühendislik metodu ile STL formatlı dosyalar, MATLAB, Solidworks makro komutları ile parça oluşum uygulamaları hakkında yapılacak çalışmalara katkı sağlamasını ümit ederim.

Yüksek Lisans eğitimim boyunca sabır gösteren aileme, tüm arkadaşlarıma ve danışman hocama teşekkür ederim.

**Balıkesir, 2023**

**Mehmet İNCE**

## 1. GİRİŞ

Günümüz koşullarında üretici firmalar ürettiği ürünlere belirli bir süre için yedek parça tedarik garantisi vermektedir. Müşteri memnuniyetine önem veren firmalar, müşterilerine kısa sürede ulaşabilmek ve hizmet kalitesini yükseltmek amacıyla yurt içi ve yurt dışında teknik servis bulundurmaktadırlar. Teknik servisler arızalı ve/veya hasarlı bir parça tespit ettiklerinde ihtiyaç duydukları bu parçayı üretici firmadan temin etmeye çalışırlar. Ürün yelpazesi çok fazla olan ve üretim sayıları yüksek olan üreticiler için yedek parça çoğu zaman stoklama, depo alanı problemi, ekstra iş gücü olarak karşılına çıkar. Ürün yelpazesi ve üretim sayıları az olan firmalar açısından ise, yedek parça taleplerini öngörmelerde zorlanarak arz konusunda problemler yaşanmaktadır.

Bu gibi problemler göz önüne alındığında üretimi yapılmayan ancak ihtiyaç duyulan arızalı/hasarlı parçaların tersine mühendislik metodu kullanılarak üretimi yapılabilir.

Gerekli parçanın STL formatlı dosyası kullanılarak üç boyutlu yazıcılarla imal edilmesi mümkündür. Aynı zamanda üretici firmalar içinde tüm yedek parçaların STL formatlı dosyalarını da internet sayfalarında paylaşmaları da söz konusu olabilir. STL formatlı dosyaların boyutları çizim formatlarına göre çok küçük olduğundan, binlerce parça olsa dahi internet sayfalarında fazla disk boyutuna ihtiyaç duyulmayacaktır. Burada şöyle bir problemle karşılaşılabilir, eğer ki parçayı üretecek olan ihtiyaç sahibi, parça üzerinde düzenleme yapmak istediğinde bu pek kolay mümkün olmayacaktır. STL formatlı verileri bir yazılım programı ile üzerinde kolayca değişiklik yapılacak formata çevrilebilir. Bu aynı zamanda üretim öncesi parçanın şeklinin de görüntülenmesine olanak sağlayacaktır.

H. Demir ve arkadaşlarının yaptığı çalışmada, geometrik yapısı itibariyle tasarım ve modelleme açısından, teknik resim çizimi olanaksız olan uyluk kemiğini, tersine mühendislik metodu ile, optik tarama sistemi Optocad programı aracılığıyla 3 boyutlu nokta yüzey formatına sonrasında da RapidForm Xor programı kullanarak 3 boyutlu CAD datası elde ederek üretimi yapılabilir hale getirmişlerdir [1].

C. Balta yaptığı çalışmada veri toplama ile cisim yüzey ve hacim bilgisini barındıran nokta bulutu verilerinin, tersine mühendislik uygulamalarında bilgisayar ortamına aktarılmasında örme yüzey ve b-spline yama şeklinde geri elde edilmesini incelemiştir [2].



M. Eragubi, STL formatındaki dosyaya kaydedilen 3B tasarım verilerini okuyan ve ardından modelin biçimini ve boyutlarını anlamlandırmak için tasarımı grafiksel olarak görüntüleyerek, modelin dijital olarak dilimlenmesi işlemi tasarımın alt kısmından başlayıp en üst noktasında bitiren program üzerinde çalışmıştır [3].

B. Duman M. C. Kayacan, üretilecek ürünün CAD programları ile tasarımdan, bilgisayarlı tomografi ile elde edilen verilerden ya da 3 Boyutlu tarayıcılardan elde edilen nokta bulutundan sonra STL dosya biçimine dönüştürülmesini belirtirken, verilerin STL dosyasına dönüşüm sırasında bazı hatalar oluşabildiğini, bu hataların onarım metotlarını karşılaştırmalı olarak analizlerini incelemişlerdir [4].

T. C. Lueth ve F. Irlinger makalelerinde, düzlemsel bir mekanizmanın bileşenlerinin geometrik katılar olarak nasıl modellenebileceğini ve 3D baskı için STL dosya formatında kaydedilebileceğini ayrıntılı olarak bahsetmişlerdir. Mekanizmaların eklemeli üretimi veya eksiksiz bir dişli çözümü için gerekli STL dosyalarının yalnızca MATLAB kullanılarak otomatik olarak nasıl oluşturulabileceğini ayrıntılı olarak sunar [5].

K. Balachandar ve arkadaşları yaptıkları çalışmada, karmaşık tasarım adımları olmaksızın girdileri verilen motor pistonu ve bağlantı çubuğunun, teknik resim çizimleri için bir uygulama geliştirmişlerdir. Solidworks API ve makro özelliği, mühendislerin CAD ve CAE'nin birçok görevini otomatikleştirmede kullanılabilecek özel programlar ve uygulamalar (makrolar) oluşturmasına olanak tanıdığından bahsetmişlerdir. Makro, tek bir düğmeye basarak standart bir çizim sayfası oluşturmaktan karmaşık tasarım özellikleri oluşturmaya kadar çeşitli görevlerin yapılabileceğini irdelemişlerdir [6].

2. Bölümde ise Tersine mühendislik metodu incelenmiş, 3. Bölümde STL formatlı dosyalar irdelenmiştir. 4. Bölümde STL formatlı dosyaların MATLAB programı ile şeklin oluşturulmasına yer verilmiştir. 5. Bölümde Solidworks programında Api komutları ile uygulamalar yapılmıştır. 6. Bölümde önceki bölümler ışığında araştırma ve çalışmamızın değerlendirme ve sonuçları sunulmaktadır.

## 2. TERSİNE MÜHENDİSLİK METODU

Tersine mühendislik, bir aygıtın, işlemin, sistemin veya bir yazılım parçasının bir görevi nasıl yerine getirdiğini, tam olarak nasıl yaptığına dair çok az iç görü ile akıl yürütme yoluyla anlamaya çalışan, uygulama dayalı bir süreç veya yöntemdir. Tersine mühendislik uygulaması, fiziksel nesnelerin dijital geometrik modellerinden yeniden inşa edilmesidir. Tersine Mühendislik uygulamaları Bilgisayar destekli tasarım alanında son yıllarda kapsamlı bir şekilde çalışılmıştır. Günümüzde çok hızlı bir şekilde gelişim göstermektedir. Tersine mühendislik, işlevsel nihai üründen değil, ürün döngüsünün herhangi bir aşamasından gerçekleştirilebilir.

Çeşitli alanlarda tersine mühendislik yapmanın birçok nedeni olabilmektedir. Tersine mühendisliğin kökeni, ticari veya askeri üstünlük sağlamak için mekanik parça analizine uzanmaktadır. İlk üretimlerinde yer alan imalat özellikleri hakkında çok az bilgiye sahip olunan veya hiç ek bilgisi bulunmayan ürünlerden tasarım özelliklerini çıkarmak için uygulanan bir analizdir [7].

Tersine mühendislik uygulamalarına ihtiyaç duyulmasının nedenlerini şöyle sıralayabiliriz:

- Üreticinin uzun zaman önce ürettiği parçayı yeniden üretmek istememesi
- Orijinal tasarımının yetersiz veya eksik bilgiye sahip olması
- Bir ürünün orijinal üreticisinin artık bulunmaması ancak ürüne talep olması
- CAD verisi olmayan veya verileri eskimiş veya kaybolmuş bir parçayı yenilemek veya üretmek için veri oluşturma.
- Fabrikasyon üretimi olan bir parçayı CAD dokümanı ile ya da standart özelliğiyle, muayene/kalite kontrol amaçlı karşılaştırılması
- Bir ürünün bazı kötü özelliklerinin ortadan kaldırılma arzusu
- Oyunlarda ve filmlerde animasyon için bir model veya heykelden 3 boyutlu veri oluşturulması
- Mimari ve inşaat dokümantasyonu ve ölçümü için
- Bir ürünün bazı kötü özelliklerinin ortadan kaldırılma arzusu
- Doku mühendisliği ile üretilmiş vücut parçaları, diş veya cerrahi protezler, oluşturmak veya cerrahi planlama için veri oluşturmak
- Suç mahallerinin belgelenmesi ve çoğaltılması.

- Rakip ürünlerin iyi ve kötü özelliklerini analiz edilmesi
  - Ürünün ilk üretici firmasının ek parça sağlama konusunda isteksiz veya yetersiz oluşu
  - Orijinal üreticinin yeni parça üretiminde çok yüksek fiyat talep etmesi
- [1], [8].

Bunlarla birlikte, sayısının artırılması veya değiştirilmesi istenen parçalara ait bilgisayar destekli tasarım modelleri bulunmaması ya da hali hazırdaki modeller ile bu parçaların üretiminin problemlili olması da tersine mühendislik metodunun uygulanmasına ihtiyaç duyurabilmektedir [9].

Piyasada bulunan parçaların daha hızlı dijital ortama aktarılması için geliştirilen yöntemler tersine mühendislik metoduna olan ilgiyi oldukça arttırmaktadır. Yazılım mühendisliği, mekatronik mühendisliği, eğlence sektörü, eczacılık, kozmetik ürün gelişimi, dental, biyomedikal, savunma sanayi, mikroişlemciler, kalıp hazırlama, otomotiv yedek parça, uzay teknolojisi, endüstriyel alanlar, arkeoloji gibi birçok alanda da tersine mühendislik yöntemleri kullanılmaktadır.

Tersine mühendislik uygulamalarını Veri toplama aşaması, Ön İşlemler: örme yüzey elde edilmesi, Segmentasyon: bölgelere ayırma, Yüzey Eşleme, Nesne Oluşturma aşamaları olarak özetlenebilir [2].



**Şekil 2.1:** Tersine mühendislik uygulama aşamaları.

Veri toplama ve örme yüzey elde edilmesi sadece dışardan alınan verilerle elde edilebilmektedir.

Yüzey geometrisini tanımlayan nokta bulutları oluşturarak parça geometrisini taramak için üç boyutlu tarayıcılar kullanılır. Bu tarama cihazları, özel araçlar olarak veya mevcut bilgisayar sayısal kontrollü makine araçlarına eklentiler olarak mevcuttur. Doğru tarama tekniğini seçmek, taranacak parçayı hazırlamak ve parçanın basamaklar, yuvalar, cepler ve

delikler gibi tüm geometrik özelliklerini tanımlayan bilgileri yakalamak için önem arz etmektedir [8].

Üç boyutlu tarama teknolojisi, bir üç boyutlu dijital model oluşturmak için şekli ve bazen görünümü (renk, doku) hakkında veri toplayarak bir nesneyi analiz etmek için bilimsel bilginin uygulanmasıdır. Bu tür analizleri gerçekleştirmek için kullanılan cihaz bir 3 boyutlu tarayıcıdır. 3 Boyutlu tarayıcı cihazları kameralara çok benzer: konik bir görsel alana sahiptirler ve görünür yüzeyler hakkında bilgi toplayabilirler. Bir ayrıntı onları farklı kılar: kamera görüş alanı içinde renk ve yüzey bilgilerini toplarken (görüntü oluştururken), 3 Boyutlu tarayıcı elde edilen görüntüleri 3 Boyutlu verileri çıkarmak için kullanır (görüş alanı içindeki mesafe ve yüzey hakkında bilgi toplar). Genellikle nesne yüzeyinin geometrik bir nokta bulutunu oluşturur. 3 Boyutlu tarayıcı tarafından üretilen görüntü, her nokta için yüzeye olan mesafeyi gösterir. Bu, yüzeyin her noktası için üç boyutlu konumu sağlar [10].

3 Boyutlu baskı olarak da bilinen eklemeli üretim, bir nesne oluşturmak için malzemeyi katman katman ekleyerek çalışan bir imalat yöntemidir. Bu malzeme ekleme, ekstrüzyon, kaynaklama, sertleştirme, mürekkep püskürtmeli biriktirme ve diğerleri dahil olmak üzere çeşitli yollarla gerçekleştirilebilir. Süreç, dijital olarak katmanlar halinde dilimlenen bilgisayar destekli tasarım (CAD) ile yapılan dijital bir 3 boyutlu tasarımla başlar. Eklemeli imalat makinesi daha sonra bu katmanların her birini oluşturmak için bir takım yolu belirler ve baskı başladığında, yazıcı parça tamamlanana kadar her katman için bu takım yolunu takip eder. Parçaları eklemeli imalat yoluyla katman katman üretmenin avantajları arasında yüksek düzeyde tasarım özgürlüğü, tek tip teslim süresi, hammadde geri dönüşümü yoluyla malzeme atıklarının azaltılması ve diğer avantajların yanı sıra tek bir baskıda çeşitli geometriler oluşturma yeteneği yer alır [11].

Eklemeli imalat yöntemlerinden olan 3 boyutlu (3B) yazıcılar, hızlı prototipleme teknolojisi olarak bilinmektedir. Karmaşık yapılarda dikkate değer avantajları olan, işlenmesi zor olan nesnelere üst üste katmanlar halinde birleştirilerek oluşturmak için polimer, kompozit, plastik gibi malzemeler kullanan, girdi verisi dijital olan teknolojidir. Endüstriyel alandan ev tipi kullanımlara kadar 3B yazıcılar geniş yelpazede kullanılmaktadır.

3B yazıcılar, modellemesi yapılmış bilgisayar destekli tasarımların prototip veya kullanılacak nihai parçalarının hızlı bir şekilde üretilmesini sağlamaktadır. 3B yazıcılarda

sıklıkla kullanılan Eriterek Yığılma Modeli, Stereolithography, Dijital Işıık İşleme, Seçici Lazer Sinterleme teknolojileridir. Eriterek Yığılma Modelinin çalışma prensibi, eritilen malzemeyi ip şeklinde akıtarak oluşturmaktadır. Stereolithography, foto polimer reçineyi lazer ile katılaştırır. Dijital Işıık İşleme foto polimer reçineyi ışık kaynağı ile katılaştırma yöntemini ile çalışır. Seçici Lazer Sinterlemenin çalışma prensibi ise Toz poliamidi lazer ile katılaştırır [12].

Parça üretmek isteyen üretici, kullanmak istediğı malzeme özelliklerini ve oluşturmak istediğı parçasına en uygun 3B yazıcı teknolojisini seçmeleri gerekmektedir.

### 3. STL FORMATLI DOSYALAR

Kullanıcılar bilgisayar destekli tasarım programlarını kullanarak yeni bir model oluşturabilir veya daha önce fotoğraflanmış ya da taranmış modelleri programlarına yükleyebilmektedirler. STL (stereo-litografi), 3D baskı ve CAD endüstrilerinde yaygın olarak kullanılan stereolitografi CAD yazılımına özgü bir dosya formatıdır. Bu programlarda tasarlanmış olan nesnelere STL uzantılı dosyalara olarak dışa aktarım yapılabilmektedirler. STL dosya formatı bilgisayar destekli tasarım formatı ile yüksek çözünürlüklü 3B yazdırma arabirimi için standardı haline gelmiştir. STL formatlı dosyalar, yalnızca üç boyutlu bir nesnenin yüzey geometrisini tanımlarken, ürünün malzeme özellikleri, renk, doku veya CAD modeli özelliklerini barındırmaz. STL dosya formatı modelin geometrik yapısını tanımlayan nokta bulutlarından oluşan üçgen yüzeylerden oluşmaktadır.

STL uzantılı dosyalarda modelin üzerinde birçok nokta oluşturularak modelin yüzey geometrisi bu noktaların oluşturduğu üçgen kümeleri ile tasvir edilmektedir. Bu üçgen kümelerinin her biri için yönlerini belirleyen bir de normali vardır. Üçgen kümelerinin her noktası x, y, z eksenlerine ait 3 adet koordinat bilgisi ile tanımlanmaktadır. Her üçgen kümesinin yüzey normaleri de x, y, z eksenlerine ait 3 adet koordinat bilgisi içermektedir. Dolayısıyla her bir üçgen 12 adet nokta koordinatı ile tanımlanmaktadır [12].

STL dosyaları ölçek bilgisi içermez ve birimler isteğe bağlıdır. Modellerin üçgen kümelerinin koordinat bilgileri ASCII ve Binary kodlama türleri olarak iki farklı biçimde tutulmaktadır. Herhangi bir metin düzenleyiciyle (ör. Notepad) açılan bir ASCII (binary olmayan) STL dosyası içinde analiz yaparsak, 3B modeli temsil eden verilerin nasıl yapılandırıldığını kolayca görebiliriz.

```
facet normal ni nj nk
  outer loop
    vertex v1x v1y v1z
    vertex v2x v2y v2z
    vertex v3x v3y v3z
  endloop
endfacet
```

Şekil 3.1: ASCII kodlama.

Binary kodlamalı STL dosyalarında 80 byte başlık bilgisine ayrılmıştır, diğer her veri için 4 byte ayrılmaktadır. Dosyadaki üçgen yüzey sayısını gösteren veri de mevcuttur. Akabinde sırayla üçgeni tanımlayan veriler izler, dosya, son üçgenden sonra biter.

UINT8[80] – Header	-	80 bytes
UINT32 – Number of triangles	-	4 bytes
foreach triangle	-	50 bytes:
REAL32[3] – Normal vector	-	12 bytes
REAL32[3] – Vertex 1	-	12 bytes
REAL32[3] – Vertex 2	-	12 bytes
REAL32[3] – Vertex 3	-	12 bytes
UINT16 – Attribute byte count	-	2 bytes
end		

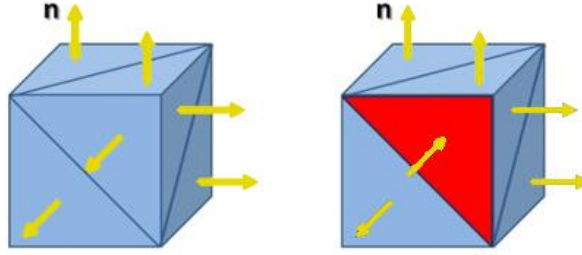
**Şekil 3.2:** Binary kodlama.

Tasarlanmak istenen modelin üçgen sayısının az olması, oluşturulmak istenen nesnenin tasarım kalitesini düşürmekte ve orijinal halinden uzaklaşmasına sebep olmaktadır. Üçgen sayısı ne kadar fazla ise, oluşturulacak nesnenin yüzey formatı orijinal haline o denli yakın olacaktır.

STL dosyalarında ki hatalar ürünün oluşumunda hatalara neden olacaktır. Hatalı üretim ise orijinal CAD geometrisinin doğru çoğaltılmasını önlemesinin yanında zaman kaybına ve maddi zarara neden olacaktır. Bu nedenle modelin başarılı bir şekilde üretilebilmesi için STL dosyasının doğruluğunun kontrol edilmesi ve dosya hataları giderilmesi gerekmektedir. Bilgisayar destekli tasarım yazılımı ile modelin STL dosya oluşturulmasında veya aktarım sırasında üçgen kümelerin yüzeyleri arasında boşluklar, kötü kenarlar, özdeş üçgenler, çakışan yüzeyler, üst üste binme, kesişme, kıvrımlar-kırıksıklıklar, içe dönük normaller gibi hatalar oluşabilmektedir.

Bir diğer önemli husus, normal vektör birimlerinin oryantasyonunun STL dosyasındaki her üçgen için nasıl tanımlandığıdır. Tüm üçgenler, nesnenin dışına doğru normal bir noktaya sahip olmalıdır. STL dosyasında tanımlanan model tersine çevrilmiş bir normal içeriyorsa

(yani ters yönü gösteriyorsa), 3 boyutlu yazıcılar da dahil olmak üzere dosyayı okuyacak olanlar artık hangisinin iç, hangisinin nesnenin dış parçası olduğunu belirleyemez [13].



**Şekil 3.3:** İçe dönük normal hatası.

Yüzeyi oluşturan üçgenlerin tüm kenarları birbirine uygun şekilde bağlanmalıdır. Aksi takdirde, delikler oluşabilir, yüzeyler üst üste gelebilir veya kesilebilir. Bu durumda, bu kenarlar "bozuk kenarlar" olarak adlandırılır. Bir STL dosyasında birbiriyle kesişen ve böylece birbirini kesen üçgenler olabilir.



**Şekil 3.4:** Bozuk kenarlar.

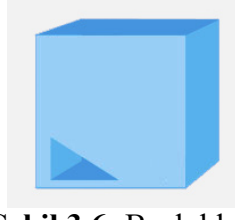
Üçgenin bir kenarı ikiden fazla yüz tarafından paylaşılır. Bu durumda, yazıcı üretim yolunu hesaplamakta zorlanacaktır. Makine ne yapması gerektiği konusunda net değildir. İki üçgen arası mesafenin küçük olması, ya da üçgenlerin normalleri arasındaki açının olması gerekenden daha küçük olması durumunda iki üçgenin çakıştığı varsayılır [2].



**Şekil 3.5:** Çakışma.

Bazen üçgenler eksiktir ve tasarımda boşluk oluşur. Bu boşluklar, parçayı düzgün bir şekilde yazdırılamamasına neden olacaktır. Delik doldurma aracı ile üçgenler oluşturularak bu hata giderilebilir [13].





**Şekil 3.6:** Boşluklar.

STL dosyalarında belirtilen hataların olup olmadığı taranarak tespit edilir. Eğer hata veya hatalara rastlanırsa onarım işlemine geçilir. Bilgisayara destekli tasarım dosyalarında, radyal taban fonksiyonları ile yeniden üçgenlerle örme ve örgü tamiri için özel yaklaşımlar tasarlanabilmektedir. Boşluklar, terslenen normaller, üst üste binmeler ve kesişme hataları gibi geometrik hataların tamirinde iyi sonuçlar alınabilecek algoritmalar oluşturulabilir.

Değişik yöntemlerle STL dosya hatalarının önemli bir kısmı giderilebilmektedir. Onarım yöntemlerinin yüzey odaklı olması, çok sağlam bir yapıya ihtiyaç yoksa ve giriş modelindeki hata sayısı az ise kullanılması doğru tercih olacaktır. Sağlam yapı olması önemli bir kıstas ise hacimsel metotlar seçilmesi daha uygun olacaktır [4].

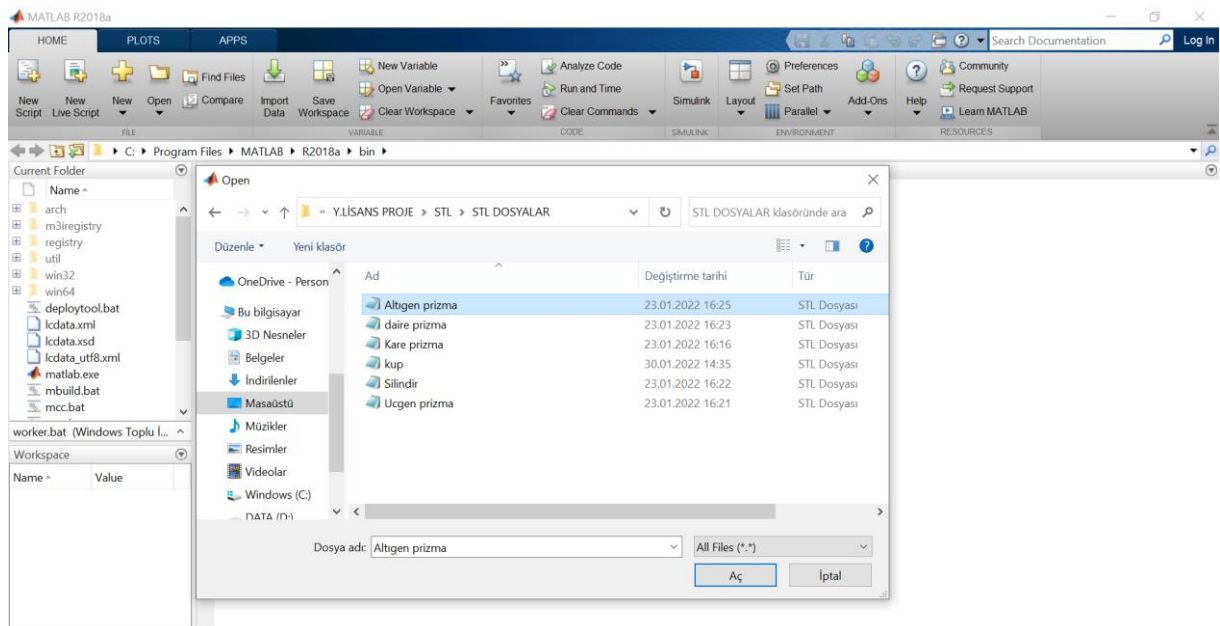
## 4. STL FORMATLI DOSYALARIN MATLABDA UYGULANMASI

Çalışmamızın bu bölümünde oluşturmak istediğimiz parçanın STL uzantılı dosyasından MATLAB programı üzerinde, yazılım geliştirerek çizdirilmesini sağlayacağız. MATLAB programı matematiksel olarak çalıştırdığından ve çizim hiyerarşisi görüntülenebildiğinden dolayı, STL dosyasından oluşacak çizim üzerinde kullanıcının istediği ölçülendirme ve değişiklikleri rahatlıkla yapabilecektir. İnternet üzerinde birçok parça, şekil için STL uzantılı dosyalarını bulmak mümkün. Önceki bölümlerde STL uzantılı dosyaların iki farklı kodlama türü olduğundan bahsetmiştik. Bunlar ASCII ve Binary kodlama türleri idi. MATLAB programında geliştireceğimiz yazılımda kullanacağımız STL formatlı dosyanın ASCII kodlama türünde olanı üzerinde çalışıyoruz. Programımızda, kullanıcıların daha rahat okuyabilmesi, sadece dosya ismini yazdığında programın otomatik olarak hangi şekil olursa olsun çizdirilmesini sağlamak amacıyla ASCII kodlama türünü tercih ettik.

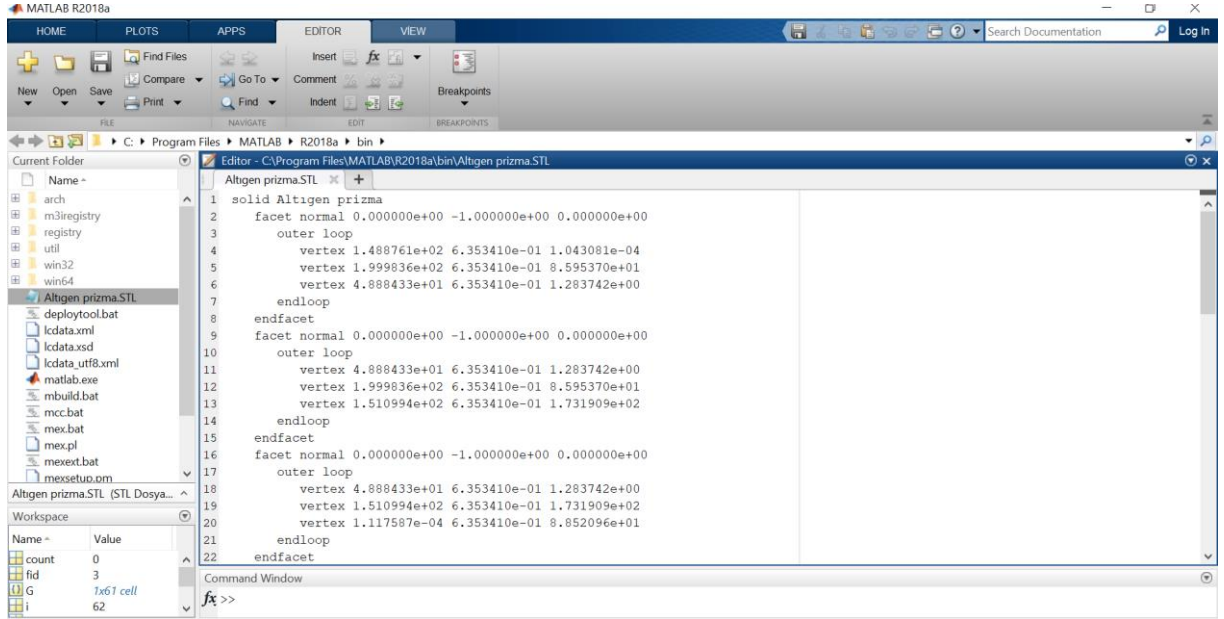
Uygulamalarımızı daha kolay anlaşılabilir olması amacıyla, çalışmamızda geometrik şekiller olan Küp, Altıgen Prizma, Üçgen Prizma, Daire Prizma, Silindir örneklerini kullanacağız.

### 4.1 Matlab Uygulamaları

Parçaların her birinin STL uzantılı dosyalarını MATLAB programına yüklenip açılması sağlanır. Çalışmamızda ilk olarak Altıgen prizma geometrik şeklini kullanacağız, dolayısıyla “Altıgenprizma.STL” dosyasını MATLAB ‘a yüklüyoruz.



Şekil 4.1: Matlab içerisinde stl dosyasının açılması, yüklenmesi.



Şekil 4.2: Matlab içerisinde açılmış olan stl dosyası.

Matlab’da yazacağımız programın her STL dosyası ile çalışabilmesi için, dosya içeriklerini ve ortak özelliklerini irdeleyelim.

ASCII kodlama türüne sahip STL dosyası satırlar halinde oluşturulmaktadır. İlk satırı her zaman, oluşturulduğu çizim programı ile ismini içeren dosya açılımla başlamaktadır.

“solid Altıgen prizma”

Sonraki satır, Örme üçgenlerinin yüzey normalinin yönünü belirtmektedir.

“facet normal 0.000000e+00 -1.000000e+00 0.000000e+00”

3. Satırda ise vertexleri oluşturan “outer loop” kalıbı yer almaktadır. 4,5,6. Satırlarda yüzey üçgenlerini oluşturacak her bir noktanın x,y,z eksenlerinde ki yerlerini belirli bir kalıp halinde ifade etmektedir.

“ outer loop  
vertex 1.488761e+02 6.353410e-01 1.043081e-04  
vertex 1.999836e+02 6.353410e-01 8.595370e+01  
vertex 4.888433e+01 6.353410e-01 1.283742e+00 “

İlk noktanın koordinatı için;

```
“ vertex v1x v1y v1z “
```

Kalıbı kullanılmaktadır. Bu durumda;

```
v1x : 1.488761e+02  
v1y : 6.353410e-01  
v1z : 1.043081e-04
```

Olmaktadır. Oluşturulacak ilk üçgenin ikinci nokta koordinatı için yine,

```
vertex v2x v2y v2z,
```

Üçüncü noktası içinde;

```
vertex v3x v3y v3z,
```

Kalıbı kullanılmıştır. Bu iki noktanın çalışmamızda ki karşılığı;

```
vertex 1.999836e+02 6.353410e-01 8.595370e+01  
vertex 4.888433e+01 6.353410e-01 1.283742e+00
```

 olduğunu görmekteyiz.

Sonraki 7. ve 8. satırlarda, “endloop” ve “endfacet” sonlandırmalarıyla ilk üçgeni oluşturacak 3 noktanın yerleri belirtilmiş olmaktadır.

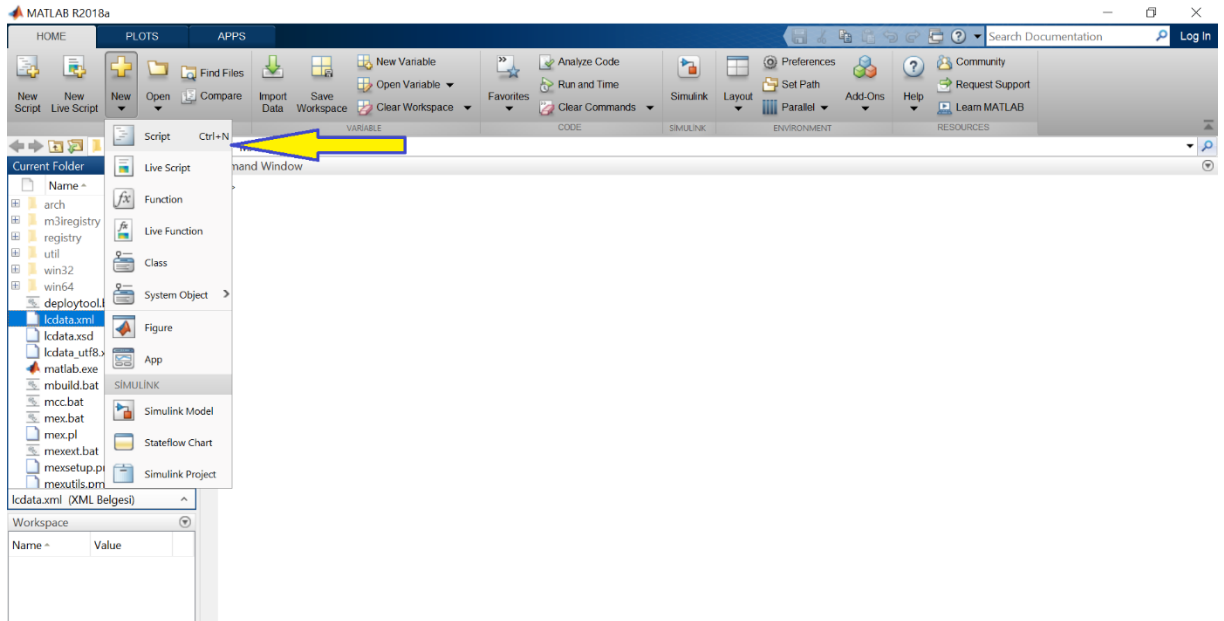
9. satırdan itibaren yeni, ikinci üçgenin yüzey normalinin yönünün tanımlanmasıyla başlamakta olup, yine ilk üçgen noktalarının gösteriminde olduğu gibi aynı satır sırasıyla, aynı kalıplar halinde nokta koordinatları belirtilmektedir.

```
“ facet normal 0.000000e+00 -1.000000e+00 0.000000e+00  
  outer loop  
    vertex 4.888433e+01 6.353410e-01 1.283742e+00  
    vertex 1.999836e+02 6.353410e-01 8.595370e+01  
    vertex 1.510994e+02 6.353410e-01 1.731909e+02  
  endloop  
endfacet “
```

STL dosyamızın içerisinde olan bütün üçgenlerin nokta koordinatları bahsedilen kalıp satır sırasıyla oluşmuştur. Ve dosyanın en son satırı ilk açılan dosya isminin kapatılması için “endsolid” kullanılarak son bulmuştur.

Bu bilgiler ışığında ilk üçgen noktamızın yeri satır ve sütun olarak belli bir noktadan başladığını görmekteyiz. Aynı şekilde ikinci ve üçüncü noktamızın da yerini satır-sütun olarak bilmekteyiz. İlk üçgenimizin ilk noktası 4. Satır 2. Sütundan başlamaktadır. Matlab programı içerisinde metin karakterlerini kaldırarak, bu nokta koordinat değerlerini okutup, bu noktalardan birbirlerine çizgi çizdirdiğimizde üçgenimizi oluşturmaktayız. Sonrasında gelen her üçgen noktaları da belirli bir satır-sütun artışıyla başladığını anlamaktayız.

Programımızı yazabilmek için önce yeni bir script oluşturuyoruz. Komutlarımızı oluşturduğumuz bu script içerisine yazarak, programımızı tamamladıktan sonra “run” komutu ile çalıştıracağız.

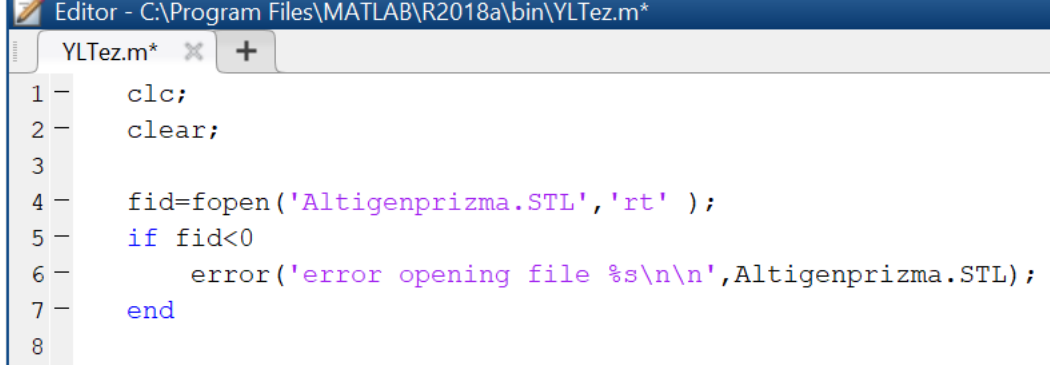


Şekil 4.3: Yeni bir script oluşturma.

Öncelikle STL dosyamızın Matlab programı tarafında açılıp okunmasını ve/veya aynı zamanda istiyorsak yapılan değişiklikleri dosya içerisine yazdırılmasını sağlamalıyız. Bunun için;

```
fid=fopen('dosya adı', 'rt')
```

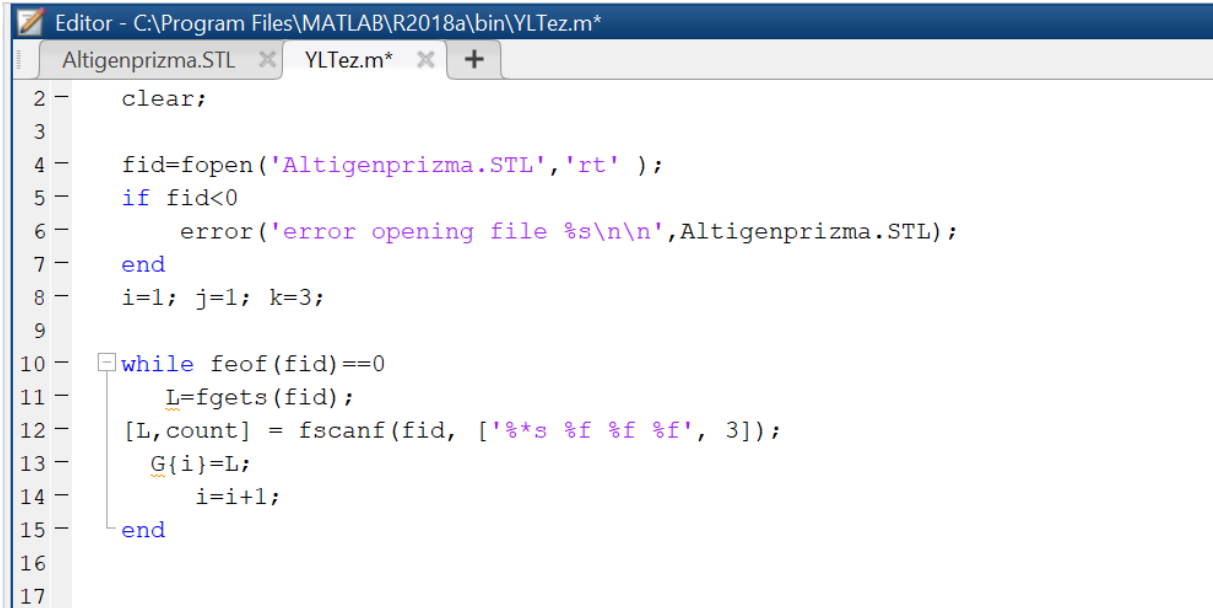
Komutunu kullanmaktayız. Eğer sadece dosya yazdırılacaksa ‘w’, dosyadan sadece okuma yapılacak ise ‘r’ kullanılmalıdır. Biz hem okuyup hem yazdırmak isteyebileceğimiz için ‘rt’ kullanıyoruz.



```
Editor - C:\Program Files\MATLAB\R2018a\bin\YLTez.m*
YLTez.m* x +
1 -   clc;
2 -   clear;
3
4 -   fid=fopen('Altigenprizma.STL','rt' );
5 -   if fid<0
6 -       error('error opening file %s\n\n',Altigenprizma.STL);
7 -   end
8
```

Şekil 4.4: Matlab’da fopen komutu ile stl dosyasının açılması.

Dosyamızdan ilk üçgenin ilk noktasının koordinatlarını almak istiyoruz. Yani 4. Satırın 2. sütunundan başlayarak 4. satırın 2., 3., ve 4. sütun değerlerini sayı olarak almak istiyoruz. Aldığımız bu değerleri öncelikle bir sayaca atıyoruz.



```
Editor - C:\Program Files\MATLAB\R2018a\bin\YLTez.m*
Altigenprizma.STL x YLTez.m* x +
2 -   clear;
3
4 -   fid=fopen('Altigenprizma.STL','rt' );
5 -   if fid<0
6 -       error('error opening file %s\n\n',Altigenprizma.STL);
7 -   end
8 -   i=1; j=1; k=3;
9
10 -  while feof(fid)==0
11 -      L=fgets(fid);
12 -      [L,count] = fscanf(fid, ['%s %f %f %f', 3]);
13 -      G{i}=L;
14 -      i=i+1;
15 -  end
16
17
```

Şekil 4.5: “fscanf” komutu kullanılarak text içerisinde ki istenilen değerlerin alınması.

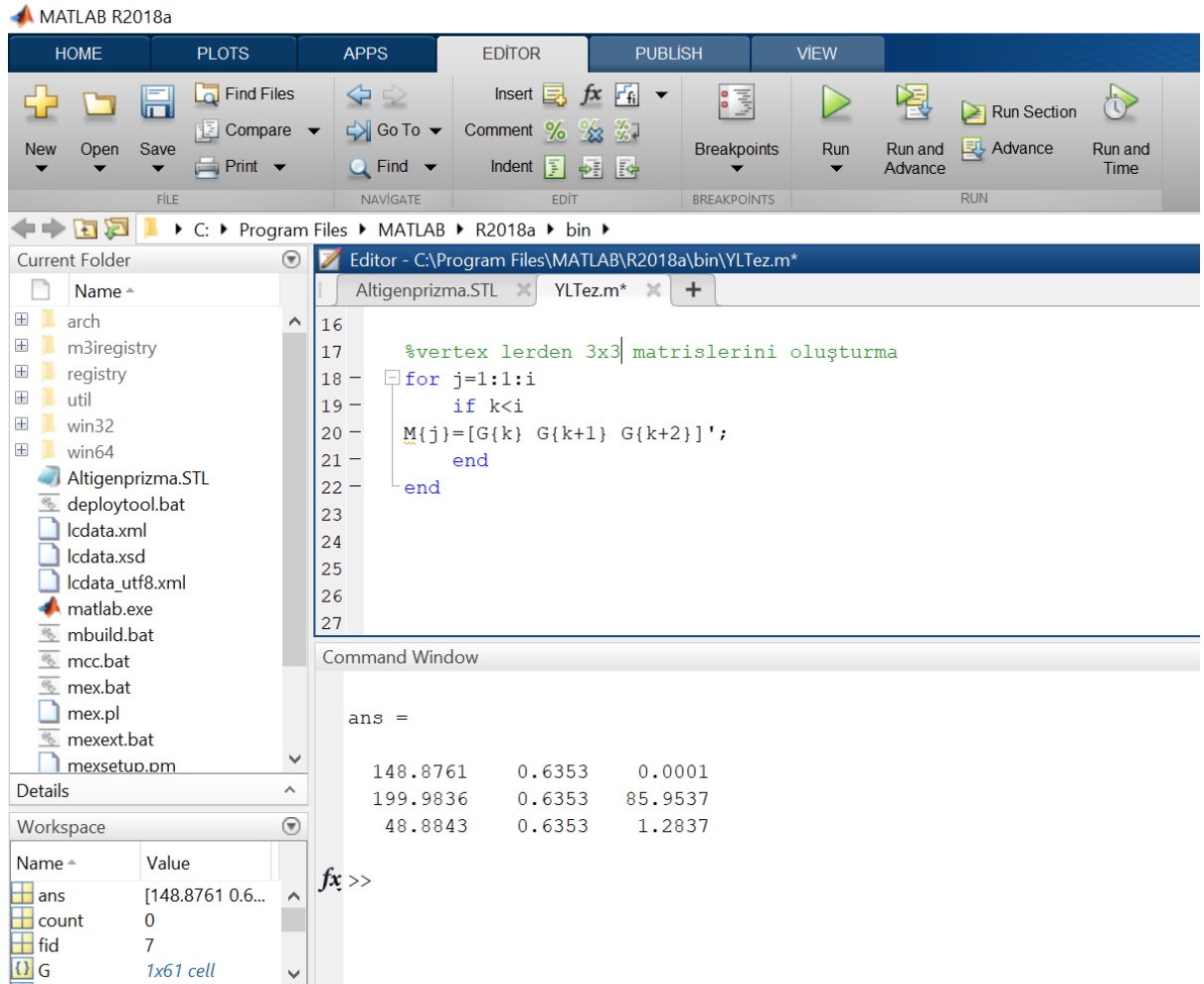
Matlab da düz yazı karakterleri string diye anılmaktadır. Her bir satır içerisinde ki sütun verilerinden alınmak istenen veri metin ise “%s” ile sayı değeri ise float anlamında olan “%f” kullanılmaktadır. Eğer ilgili sütun alınmak istemiyorsa karakter özelliğinin başına yıldız “\*” işareti konulmaktadır. Değerlerini almak istediğimiz 4. satır “vertex” ile başladığından ilk sütun değeri düz yazı şeklinde yani metindir. Sonraki gelen 3 sütun verisini almak istediğimiz sayı değerindedir. Dolayısıyla komutumuzu yazarken ilk metin değerini

almak istemiyoruz ve bunu “%\*s” olarak ifade ediyoruz. Almak istediğimiz her bir sütun değeri için sırasıyla “%f” ifadesini kullanmaktayız.

Alınan değerler L sayacına tanımlanmaktadır. Bir döngü içerisinde her bir satır için alınmak istenilen veriler sadece değer olarak bu şekilde tanımlanmaktadır. Akabinde alınan yüzey üçgenlerinin her bir nokta koordinatı olarak tanımlanan  $G\{i\}$  dizisinin değerleri olarak atanmaktadır. Yani dolayısıyla ilk üçgene ait 3 nokta koordinat değerleri,

$$G\{1\} = 1.488761e+02 \quad 6.353410e-01 \quad 1.043081e-04$$

dizi değerlerine tanımlanmış/atanmış olmaktadır. Döngü içerisinde oluşturduğumuz  $G\{i\}$  dizisindeki “i” değeri STL dosyasında ki son yüzey üçgenin son noktasının z koordinat değerine kadar devam edecektir. Bu bilgiler ışığında, ilk yüzey üçgenine ait 3 nokta koordinatlarını 3x3 lük bir  $M=[ ]$  matrisi oluşturmak için, 1 den “i” değerine kadar devam edecek şekilde bir “for” döngüsünden yararlanıyoruz.



Şekil 4.6: 3x3 lük vertex matrislerinin oluşturulması.

STL dosyasında kaç adet üçgen var ise o sayıda  $M[]$  matrisi oluşacaktır. Nokta koordinatlarından matris oluşturmakta ki amacımız bu matrisin değerlerinden birbirlerine çizgi çizdirerek üçgenleri oluşturmaktır. Bu çizgilerin çizilmesini for döngüsü içerisinde çıkmadan, her bir üçgen için  $M[]$  oluşturduğumuz anda, noktalar arası yani matris değerleri arası çizimleri de yapmaktayız. Oluşacak bir sonraki üçgenin ilk nokta koordinatı ilk üçgenin değerinden 6 karakter sonra başladığını bildiğimiz için, bir sonraki üçgen noktaları için oluşturacağımız  $M[]$  matris değerleri içeriğini de her defasında 6 arttırarak başlatmamız yeterli olacaktır.

Burada dikkat edilecek hususlardan bir tanesi de, döngü içerisinde çizim yaptırdığımız için, ilk çizimden son nokta çizimine kadar olan bütün çizimler aynı grafik üzerinde gösterilmelidir. Böylece ilk matris çizimleri tamamlandığında grafik üzerinde bir üçgen göreceğiz ve STL dosyasında ki var olan bütün değerler alınarak oluşan bütün matrislerin çizimleri tamamlandığında bize oluşacak şeklin görüntüsünü üçgen yüzeyleri oluşmuş şekilde gösterecektir.

```

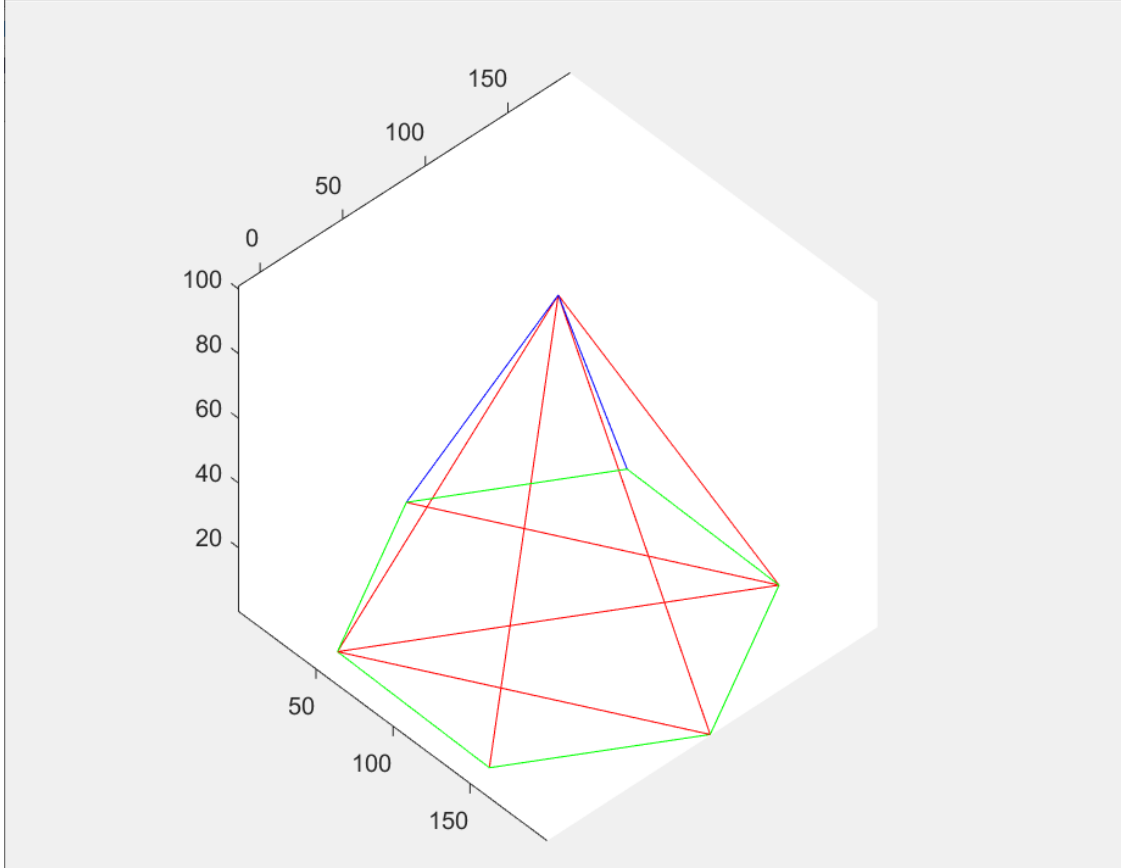
16
17 %vertex lerden 3x3 matrislerini oluşturma
18 for j=1:1:i
19     if k<i
20         M{j}=[G{k} G{k+1} G{k+2}];
21
22 % üçgenleri çizdirme
23 plot3(M{j}(1:2,1),M{j}(1:2,2),M{j}(1:2,3),'r')
24 hold all
25 plot3(M{j}(2:3,1),M{j}(2:3,2),M{j}(2:3,3),'g')
26 plot3(M{j}(1:2:3,1),M{j}(1:2:3,2),M{j}(1:2:3,3),'b')
27
28 k=k+6;
29 end
30 end
31

```

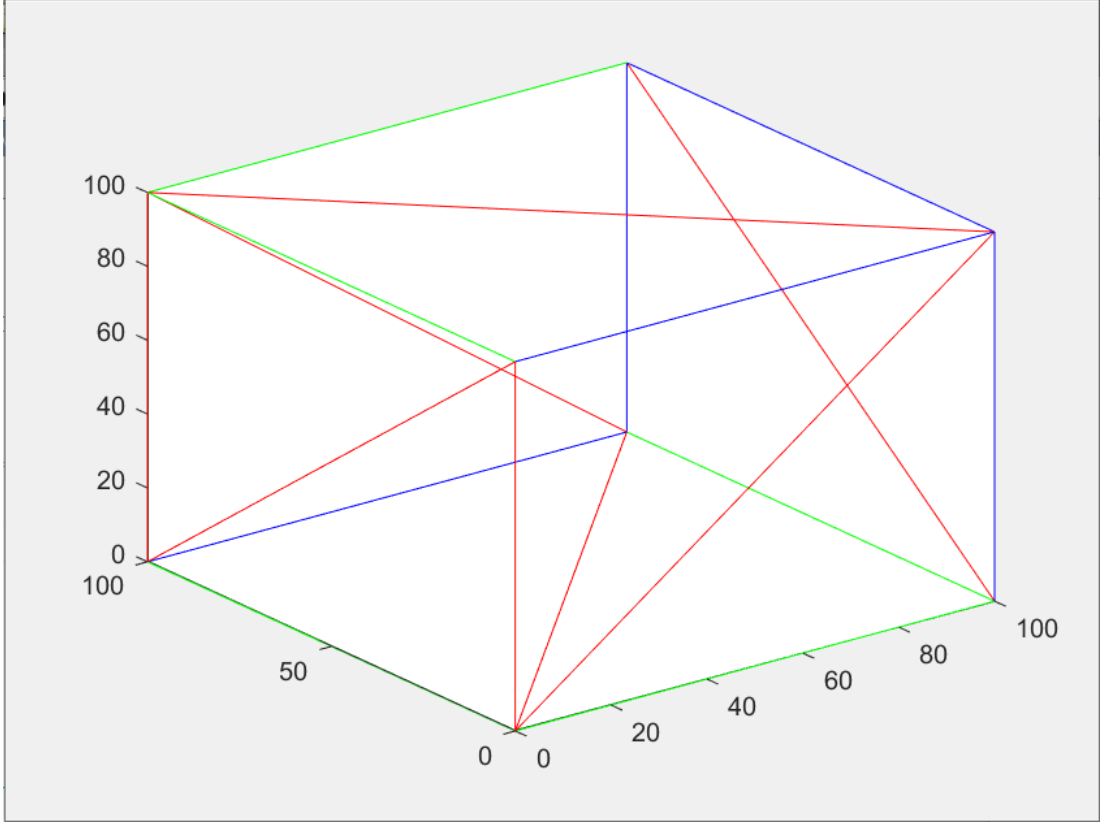
**Şekil 4.7:** Matris değerleri arasında çizgiler çizilerek üçgenlerin oluşturulması.



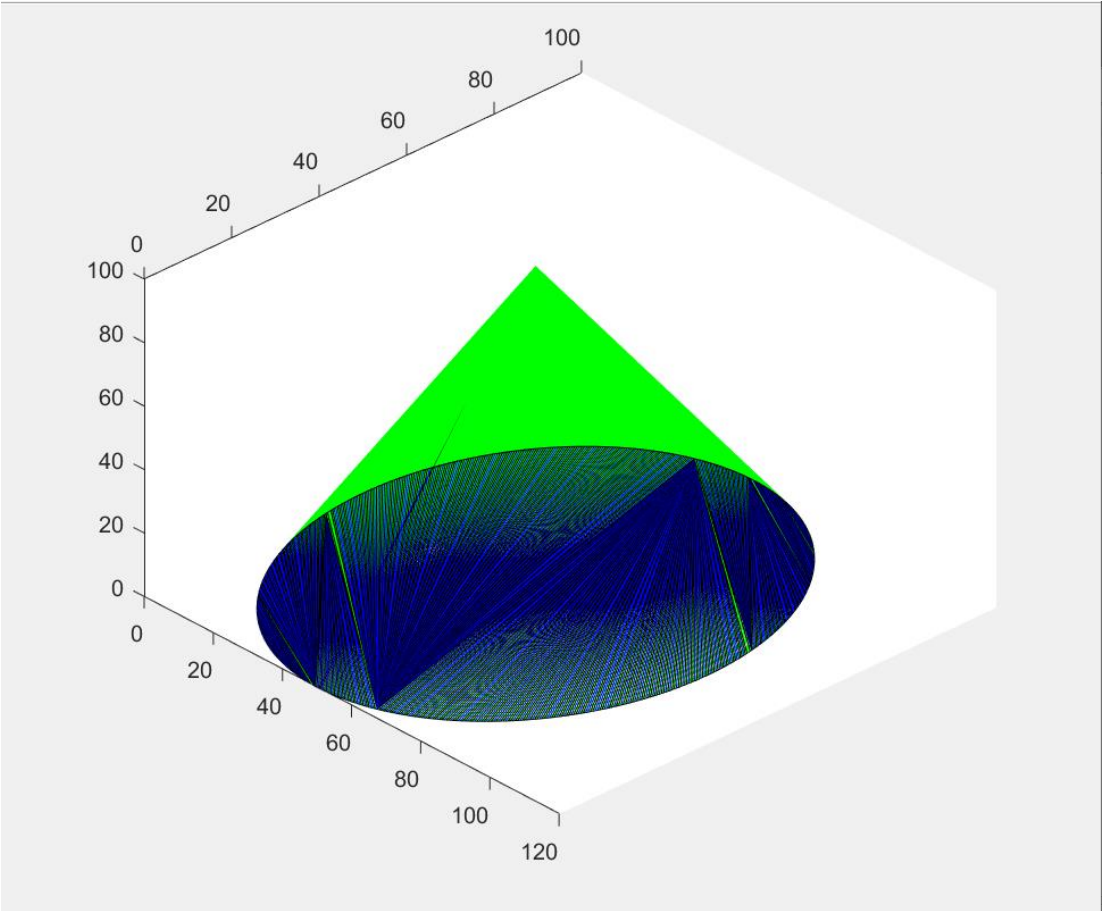
Yazdığımız komutları script açarak yazdığımız için programı çalıştırmak için Run komutunu kullanarak çalıştırıyoruz. Böylelikle, sadece oluşturulmak istenen parçanın STL dosya ismini programımızda “fopen” komutunu kullanarak açmak istediğimiz dosya ismi olarak değiştirerek yazdığımızda ve run komutuna bastığımızda, cismin şeklini bize grafikte göstermektedir. Yaptığımız geometrik şekillerin Matlab da çizdirilmiş sonuçları aşağıda gösterilmiştir.



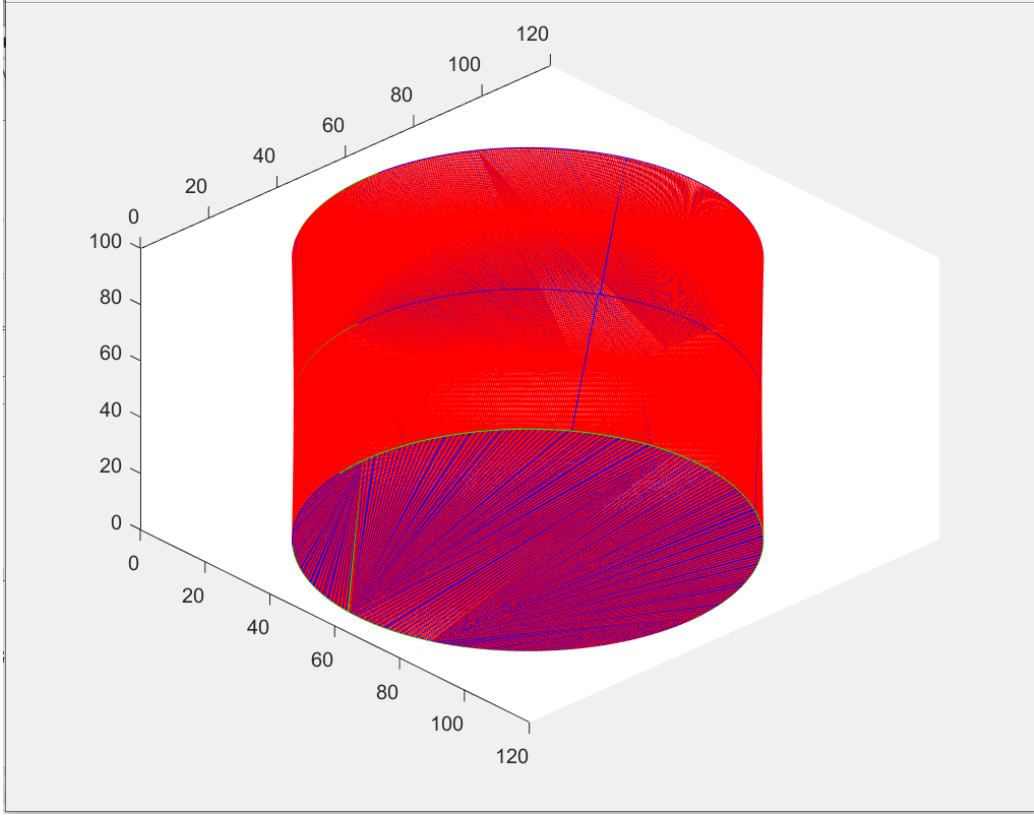
**Şekil 4.8:** Programımız çalıştırılarak çizdirilmiş altıgen prizma.



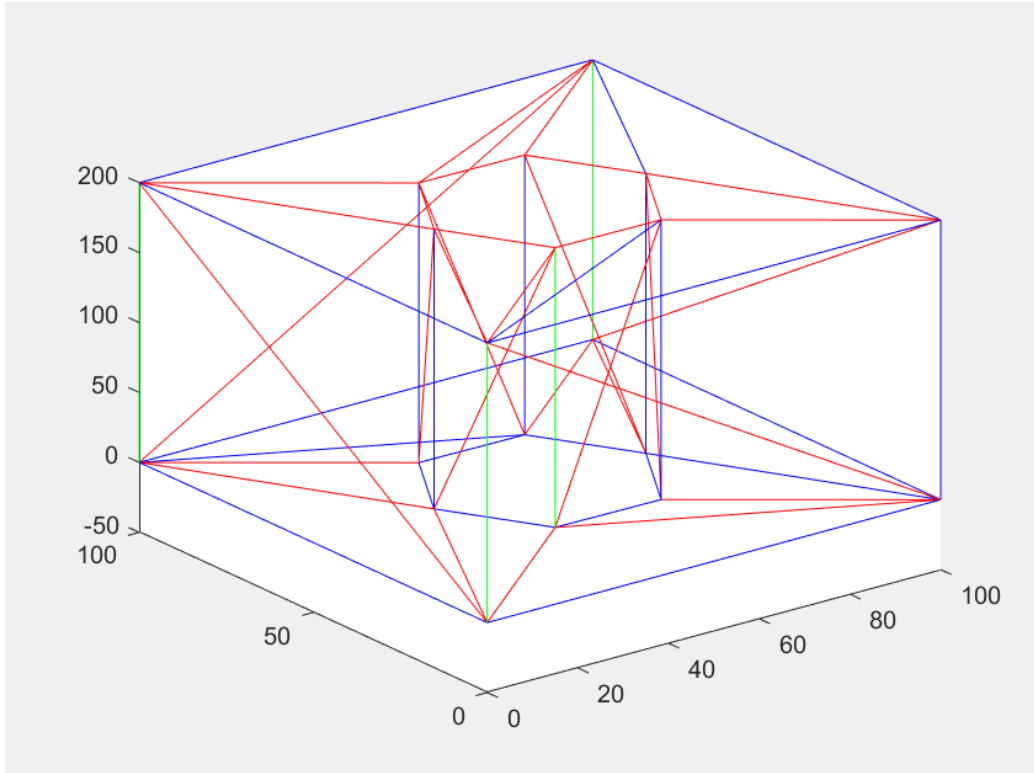
**Şekil 4.9:** Programımız çalıştırılarak çizdirilmiş küp.



**Şekil 4.10:** Programımız çalıştırılarak çizdirilmiş daire prizma.



**Şekil 4.11:** Programımız çalıştırılarak çizdirilmiş silindir.



**Şekil 4.12:** Programımız çalıştırılarak çizdirilmiş altıgen delikli dikdörtgenler prizması

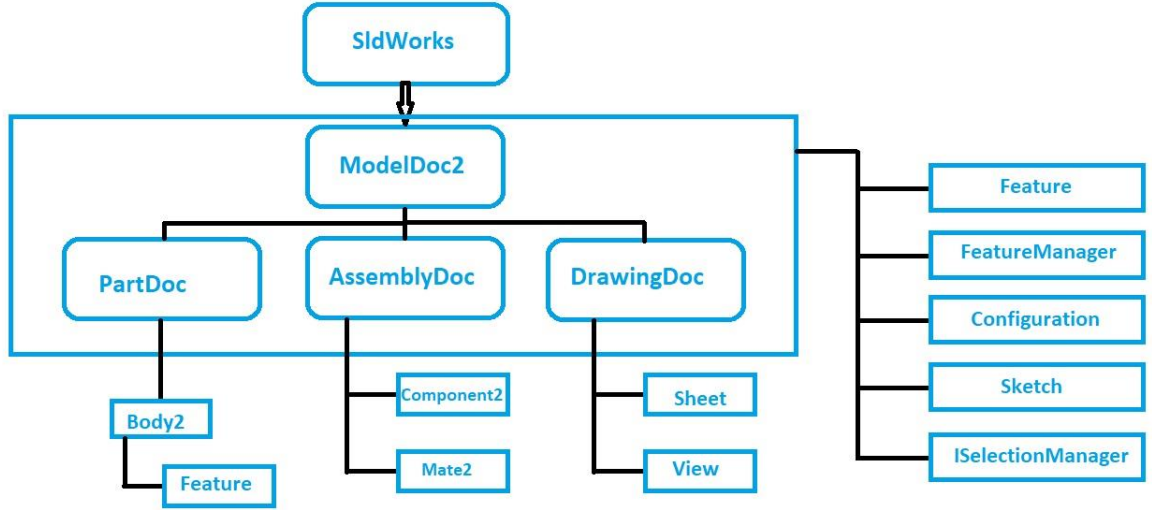
## 5. SOLIDWORKS API ARAYÜZ UYGULAMASI

İki boyutlu, üç boyutlu katı modeller ortaya çıkarmak için kolay, hızlı ve etkin bir şekilde kullanılabilen bir CAD yazılımıdır, Solidworks. Haricinde ki CAD programları gibi kendine özgü yapı oluşturarak pratiklik anlamında özgün hedeflere uygun olarak tasarlanmıştır. Anlaşılır komutlarla basit çizimlerden karmaşık üç boyutlu geometrik yapılara kadar farklı tasarımların oluşturulmasını sağlayan arayüzü, büyük ilgi görmesinin en büyük nedenlerindedir. CAD yazılımlarının sahip olmadığı üç boyutlu Sketch özelliğinin var olması Solidworks de kullanıcıya çizim yapma kolaylığı sağlıyor. Yapılan tasarımın her bir detayının görülmesini sağlaması, olası hataların erken tespit edilmesine fırsat veriyor. Katı modeli hazırlanan parça eğer gerekirse üretime geçilmeden önce yeniden boyutlandırılabilir.

Sahip olduğu Api ve makro özelliği, CAD in birçok görevini otomatikleştirme için kullanılabilecek özel programlar ve/veya uygulamalar geliştirmek isteyen mühendislere fırsat vermektedir. Bizler de çalışmamızın bu bölümünde Solidworks un Api arayüzü ile makro komutları kullanarak çizim örnekleri yapacağız. Daha bu ve benzeri çalışmalar sonrasında, Matlab programında STL dosyasından hazırlanan çizimler Solidworks Api arayüzü ile entegre edilerek, görsel imkân sağlayarak kullanıcının istediği değişiklikleri rahatlıkla yapabilmesi sağlanabilir. Yaptığımız çalışmada, Solidworks Api arayüzü makro oluşturulup, basit geometrik şekil olarak küp ve silindir çizimleri yaptırılmıştır.

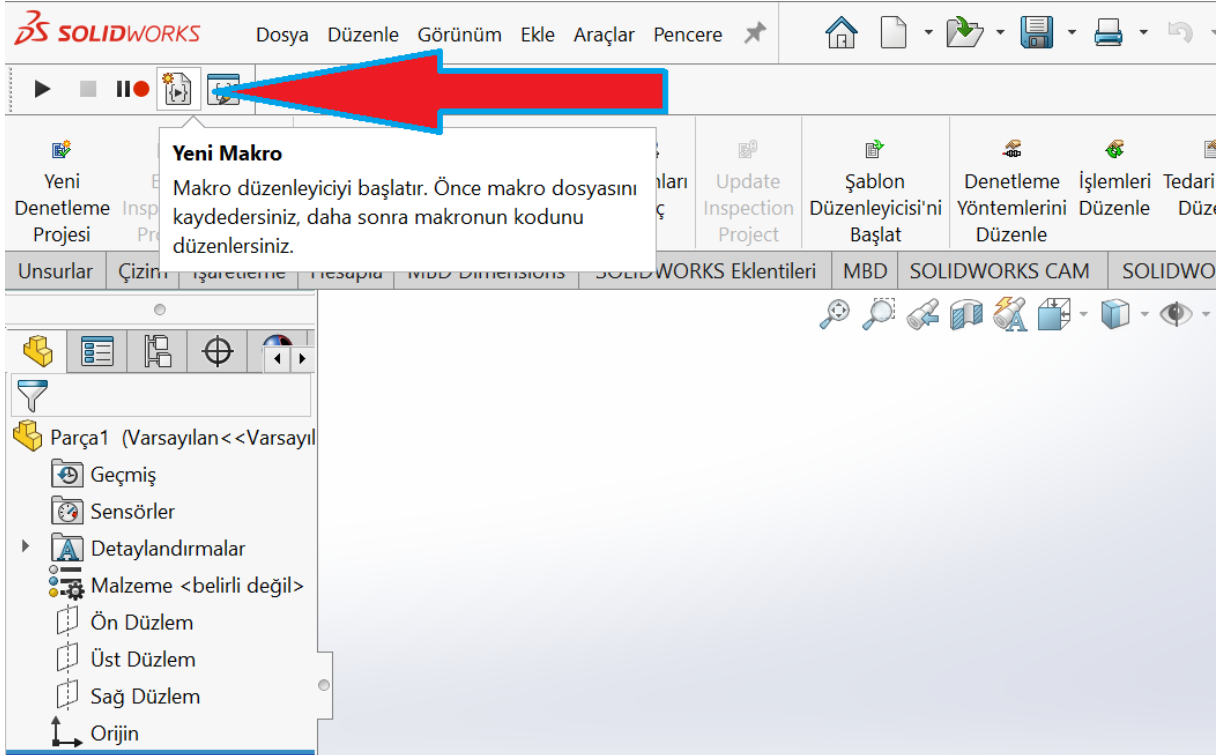
### 5.1 Solidworks Api/Makro Uygulamaları

Solidworks programında herhangi bir makro yazabilmesi için, Sldworks 'e tanımlanması gerekmektedir. Bu tanımlamayı “application.Sldwkr” komutu kullanılarak yapıyoruz ve programımızı Solidworks ile bağlantısını kurmuş oluyoruz. Akabinde yine benzer şekilde Modeldoc2 arayüzüne bağlamamız gerekiyor. Çizim için gerekli birçok komut Modeldoc2 arayüzü altında çalışmaktadır. Aşağıda gösterilen hiyerarşik sıra referans alınarak, kullanmak istediğimiz çizim komutları hangi komut kütüphanesinde çalışıyor ise, uygulamamız o arayüzde tanımlanması gerekmektedir. Örneğin Sketch açmak için Modeldoc2 arayüzünde tanımlanması yeterli olacaktır.



Şekil 5.1: Solidworks makro komutları için hiyerarşik akış sırası.

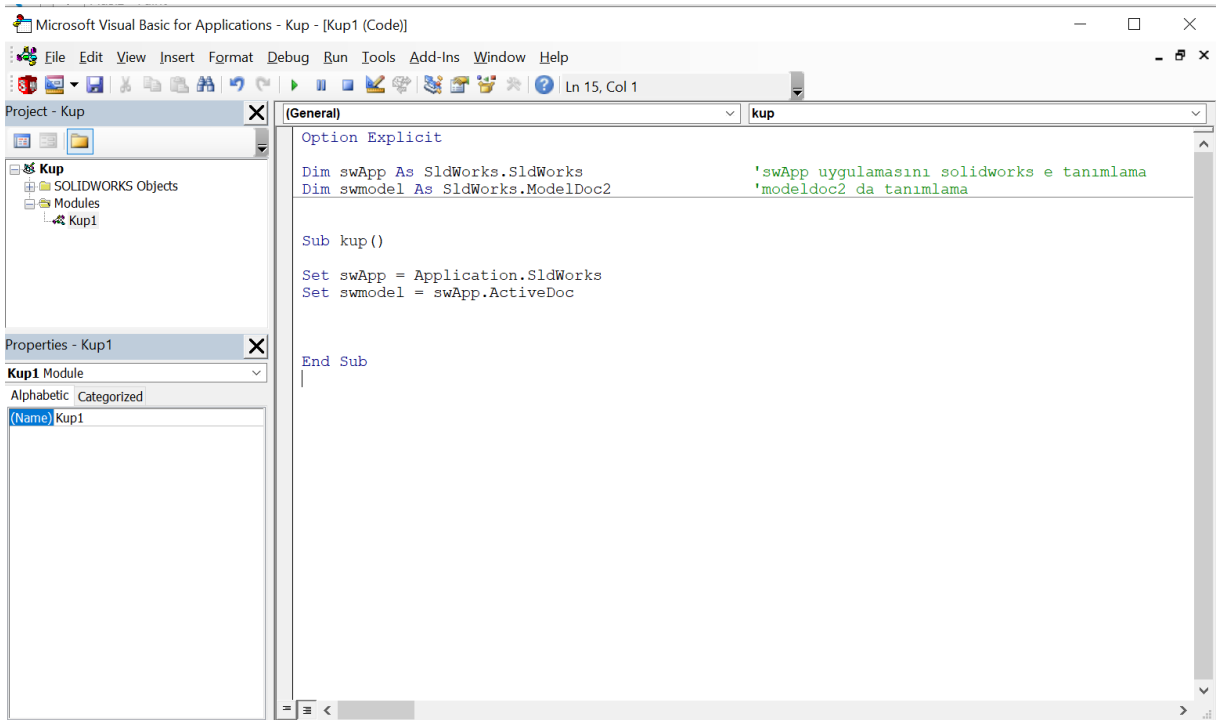
Bir makro komut dizisi oluşturmak için, öncelikle Solidworks programı açıldıktan sonra yeni belge sayfası açılarak, Makro düzenleyicisi için yeni makro arayüzü açılması gerekmektedir. Önce makro dosyasını kaydederek daha sonra makro kodlarımızı yazmaya başlıyoruz. Solidworks de hazırlayacağımız yazılımın daha iyi anlaşılabilmesi için hiyerarşik sırasının analiz edilmesi gerekmektedir.



Şekil 5.2: Solidworks programında yeni makro düzenleyicisinin oluşturulması.

Açılan makro sayfasında kullanılan komutlar Visual Basic yazılım komutları kullanılmaktadır. Öncelikle “sub main ()” dosyası açılarak komut yazılımları bu sub main dosyası içine yazılacaktır. Komut dizisi tamamlandıktan sonra “End Sub” komutu ile açılan main dosyası kapatılacaktır.

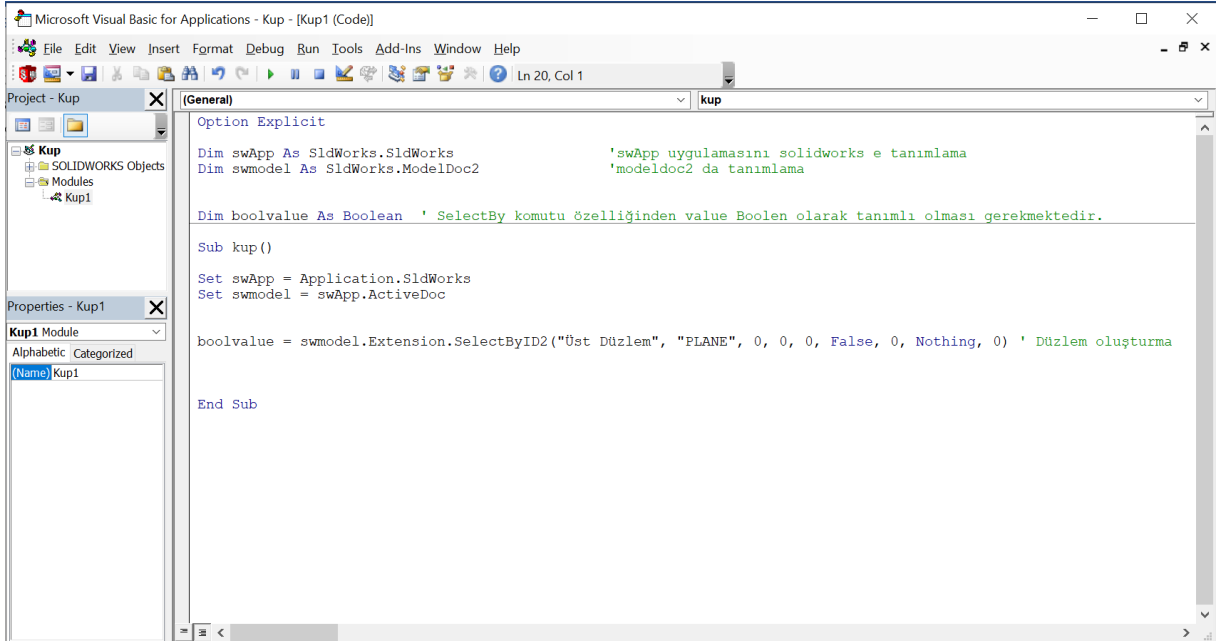
Dim komutu ile tanımlamalar yapılmaktadır. Uygulamamızı önce swApp adıyla SldWorks.SldWorks e tanımayıp, Application.SldWorks komutu ile solidworkse bağlantısını kurmuş oluyoruz. Ve akabinde swmodel adıyla uygulama modelimizi SldWorks.ModelDoc2 tanımladıktan sonra swApp.ActiveDoc komutu ile solidwrokse bağlantısını tamamlıyoruz.



Şekil 5.3: Makro düzenleyicisi içerisinde solidworks bağlantısı tanımlanması.

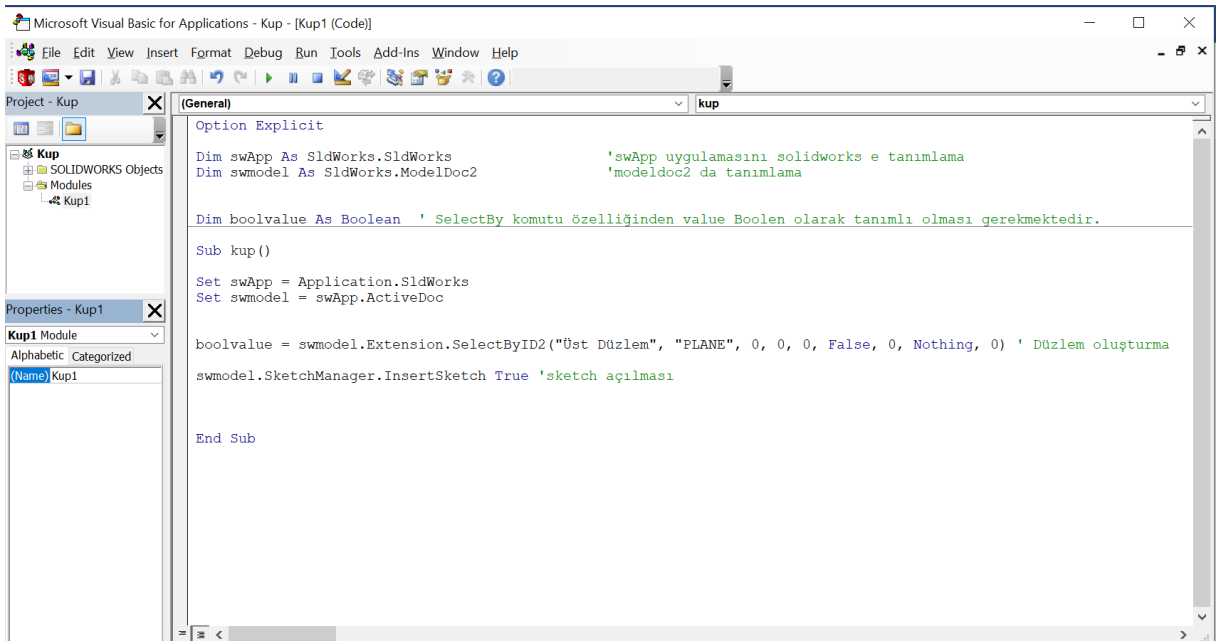
Çalışma yapacağımız düzel seçimini, oluşturulmasını SelectBy komutunu kullanmaktayız. Bu komut ModelDoc2 kütüphanesinden çağrıldığı için başka bir kütüphane tanımına ihtiyaç yoktur. SelectBy komutun değeri (value) Boolean olarak tanımlanması gerektiğinden, düzlem ataması yapacağımız değeri de Boolean olarak tanımlamamız gerekmektedir. Komut içerisine koordinat değerleri ve diğer düzlem özellikleri girilerek düzlem oluşturma tamamlanmaktadır.

Kullanıcı çizimini hangi düzlemde yapmak isterse o düzlemi seçebilir. Bunlar “Üst Düzlem”, “Ön Düzlem” veya “Sağ Düzlem” seçimi olabilir.



Şekil 5.4: Çizimin yapılması istenilen düzlemin oluşturulması.

Düzlem seçiminden sonra Solidworks özelliği olan Sketch açılması gerekmektedir. Sketch açmak için SketchManager.InsertSketch komutunu kullanılmaktadır. Komutumuzu hiyerarşi sırasından kontrol ettiğimizde ModelDoc2 da tanımlı olduğunu görüyoruz, dolayısıyla yeni bir kütüphaneye ihtiyaç duyulmamaktadır.

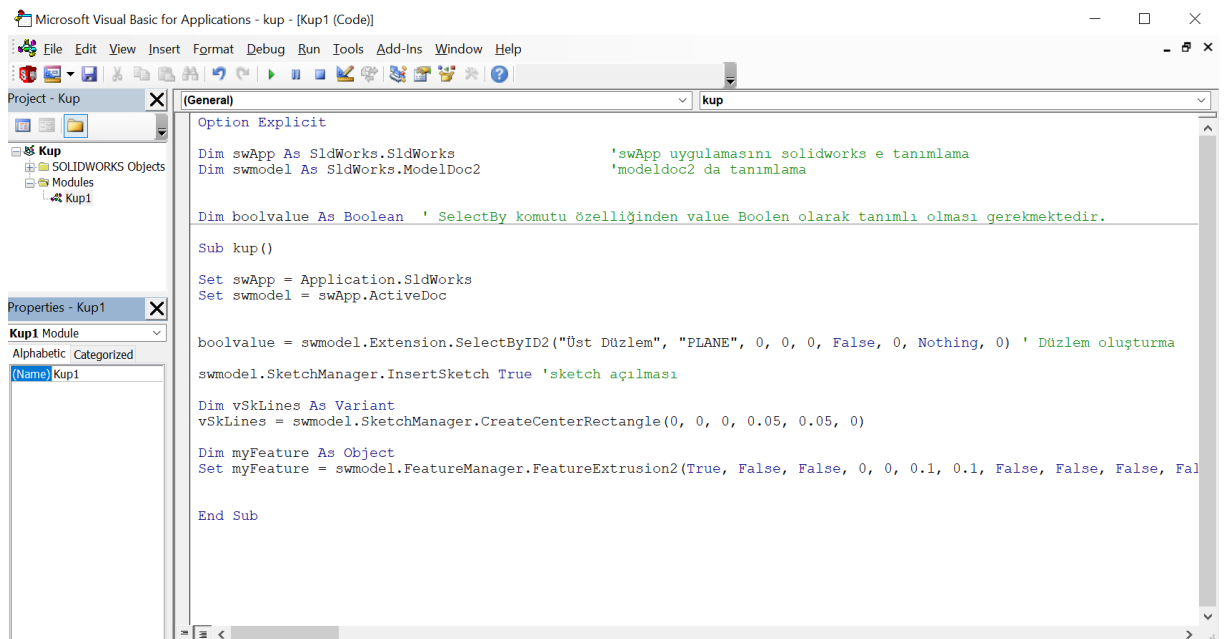


Şekil 5.5: Sketch açılması.

Sketch açıldıktan sonra artık çizilmek istenilen parçanın özelliklerine göre çizim komutlarına başlanılabilir. Çalışmamızda örnek olarak küp çizdirmek istiyoruz. Bunun için öncelikle bir kare oluşturup akabinde karenin kenar uzunluğu kadar bir katılama işlemi yaparak küp şeklini oluşturabiliriz.

Kareyi oluşturmak için dikdörtgen oluşturma komutu olan “CreateCenterRectangle” komutunu kullanmaktayız. Zira kenarları eşit olan dikdörtgenin kare olduğunu bilmekteyiz. Komutu kullanırken dikkat edeceğimiz hususlardan bir tanesi, komut içerisinde girilecek koordinat bilgisi, dikdörtgenin merkezi olmasıdır. Yine aynı şekilde, merkez olacak koordinat bilgisine olan uzaklıkları eşit değerde girersek, dikdörtgenin kenar uzunlukları da eşit olacaktır ve dolayısıyla kare oluşacaktır.

Katılaştırma için kullanacağımız komut “FeatureExtrusion” komutudur. “Feature” komutunu hiyerarşik sırasından kontrol ettiğimizde ModelDoc2 içerisinde çalıştığını görüyoruz. Tanımlamalarımızı ve komutlarımızı ModelDoc2 içerisinde kullanmamız yeterli olacaktır. Oluşturduğumuz karenin kenar uzunluğu kadar katılaştırma yapacak olursak, bir küp elde etmiş olacağız. Karenin çizimini merkez seçimli yaptığımız için, merkezi kenar uzunğunun yarısı izdüşümüne denk gelmektedir. Yani çizim için kullandığımız komutta girdiğimiz uzaklık değerinin iki katını, katılaştırma için kullanırsak tam bir küp elde etmiş oluruz.



```
Option Explicit

Dim swApp As SldWorks.SldWorks           'swApp uygulamasını solidworks e tanımlama
Dim swmodel As SldWorks.ModelDoc2       'modeldoc2 da tanımlama

Dim boolvalue As Boolean ' SelectBy komutu 8zelliğinden value Boolean olarak tanımlı olması gerekmektedir.

Sub kup()

Set swApp = Application.SldWorks
Set swmodel = swApp.ActiveDoc

boolvalue = swmodel.Extension.SelectByID("Üst Düzlem", "PLANE", 0, 0, 0, False, 0, Nothing, 0) ' Düzlem oluşturma
swmodel.SketchManager.InsertSketch True 'sketch açılması

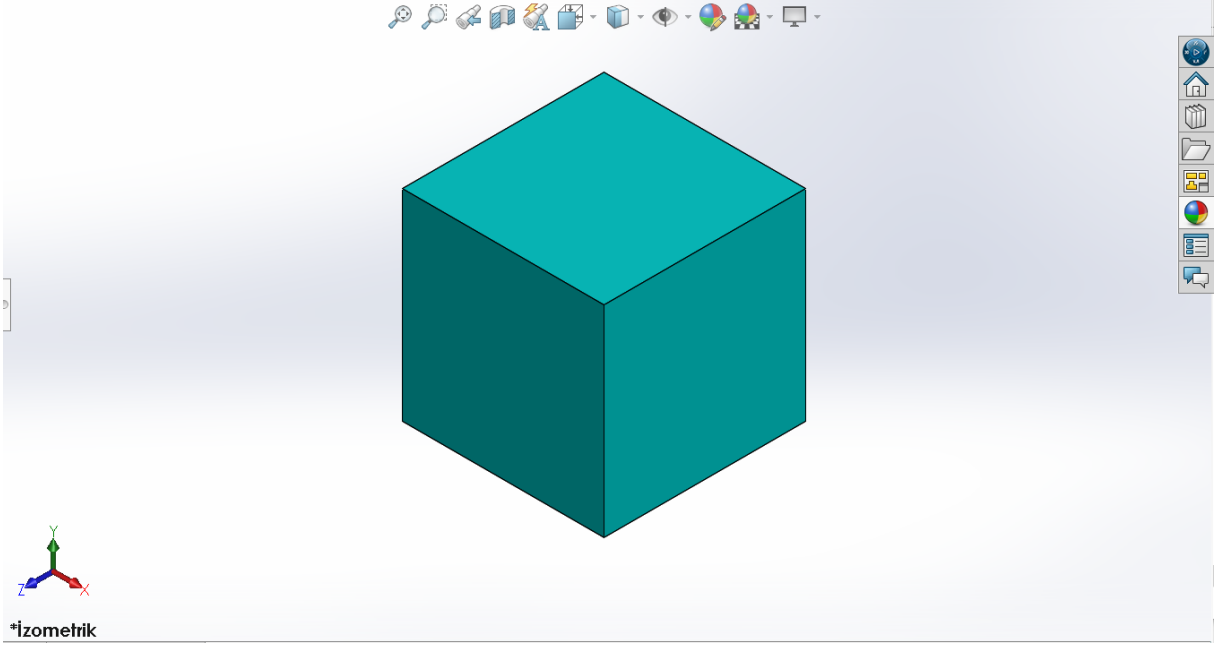
Dim vSkLines As Variant
vSkLines = swmodel.SketchManager.CreateCenterRectangle(0, 0, 0, 0.05, 0.05, 0)

Dim myFeature As Object
Set myFeature = swmodel.FeatureManager.FeatureExtrusion(True, False, False, 0, 0, 0.1, 0.1, False, False, False, Fal

End Sub
```

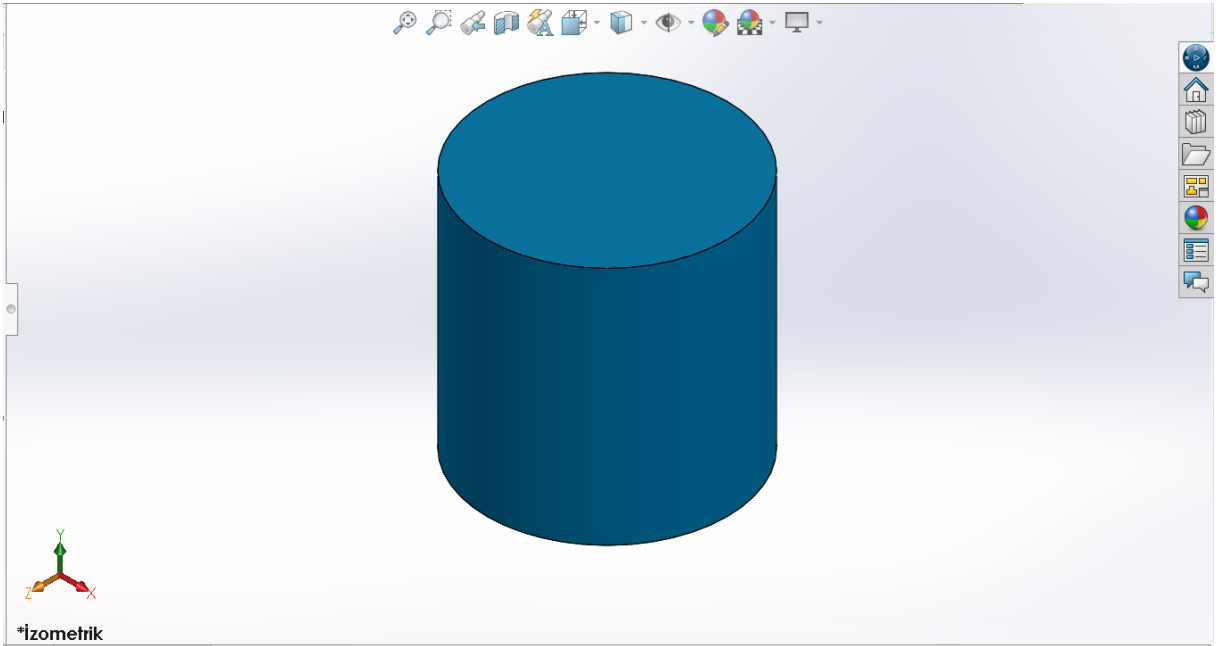
Şekil 5.6: Kare çizimi ve katılaştırma.





**Şekil 5.7:** Api arayüzü ile makro komutu çalıştırılarak çizdirilmiş bir küp.

Çalışmamıza ikinci bir örnek olarak, bir silindir oluşturmak istersek öncelikle bir daire çizdirmemiz gerekir. Daire çizimi için “CreateCircleByRadius2” komutunu kullanmaktayız. Dairenin çizdirmek istediğimiz yer bilgisi olarak, koordinat değerlerini ve yarıçap değerlerini girerek oluşturuyoruz. Akabinde yine katılaştırma için yine “FeatureExtrusion” komutu kullanmaktayız. Silindir yüksekliğimizin ne kadar olmasını istiyorsak, komut içi değerlerinden katılaştırma değerlerine girerek komutumuzu tamamlıyoruz.



**Şekil 5.8:** Api arayüzü ile makro komutu çalıştırılarak çizdirilmiş bir silindir.

## 6. DEĞERLENDİRME, SONUÇ ve GELECEK ÇALIŞMALAR

### 6.1 Değerlendirme ve Sonuç

Projemizde tersine mühendislik yöntemi ile STL formatlı CAD bilgisi hazırlanan nesnenin MATLAB gelişmiş programı üzerinden oluşturulması sağlanmıştır. Tersine mühendislik metodu ve STL dosyalar hakkında geniş literatür taramaları yapılarak çalışmamıza katkı sağlayacak çalışmalar irdelenmiştir. Üretimi olmayan ancak ihtiyaç duyulan mekanik parçaların veya hasar görmüş mekanik parçaların tersine mühendislik yöntemi ile modelleme oluşturularak nesnenin imal edilebilmesi için görsel şekli tasarlanarak, kullanıcının istediği düzenleme, ölçülendirme ve tasarım yapabilir haline getirilmiştir.

Oluşturulmak istenen parçaların şekli çizilerek, fotoğraflanarak veya taranarak STL formatlı nokta bulutlarının oluşturulması, noktaların birleşimi ile oluşan üçgen analizleri ile yüzey modellemesinin çıkarılmış CAD bilgisi ile tersine mühendislik yöntemine dayalı bir nesne oluşumu tasarlanmıştır.

STL formatlı modelinin seri katmanlı dilimlemesini gerçekleştirmek için Matlab yazılım platformları altında dosya okuma, matris, dizi, döngü gibi fonksiyonların nasıl kullanılabileceği açıklanmıştır.

Yaptığımız bu çalışma neticesinde hazırladığımız program ile verilen STL formatlı dosyadan oluşturulmak istenen karmaşık şekillerin en iyi halinin algılanması sağlanmıştır. Çalışmamızda yapılan uygulama örnekleri, temel geometrik şekiller üzerinde tasarlanmıştır ancak nihayetlenen yazılımımız, herhangi bir şeklinin ASCII kodlamaya sahip STL dosyası ile çalıştırıldığında karmaşık şekillerin dahi, nokta bulutları birleştirilerek yüzey örme üçgenleri oluşturularak parçanın şeklinin yapısını çizdirerek göstermektedir.

Çalışmamız ile oluşan yüzey üçgenlerinin sayısı ne kadar fazla ise parça şeklinin oluşumu, parçanın ham haline o denli yakın olabileceğini ortaya koymaktadır.

Solidworks programı Api arayüzü ile makro komutları kullanılarak bir parça şeklinin nasıl oluşturulabileceği açıklanmıştır. Makro komutları ile yapılan çizim çalışmaları temel geometrik şekillerin çizdirilmesi sağlanmıştır.

Çalışmamızda STL formatlı CAD bilgisi hazırlanan nesnenin, tersine mühendislik metodu ile MATLAB programı üzerinden oluşturularak, bu doğrultuda ülkemizde yapılacak çalışmalara temel fikir olarak sunulmuştur.

## **6.2 Gelecek Çalışmalar**

Gelecekte yapılacak çalışmalar ile, oluşturulan parçanın görüntülenmesine ilave olarak, şeklin geometrik özellikleri olan, benzediği en yakın geometrik şeklin adı, kaç köşeye sahip olduğu, yüzey kenar sayısı, iç bükey, dış bükey yüzeyleri var ise belirtilmesi gibi yazılımımız geliştirilerek kullanıcıya sunulabilir.

Ayrıca MATLAB ile Solidworks Api arayüzünde hazırladığımız programımız sonraki çalışmalarda birbirlerine entegre edilerek daha geniş yelpazede ve son kullanıcıya sunulan çıktı geliştirilebilir.

Çalışmamızda hazırlanan programın son hali, kullanıcıya düzenleme yapmak isteyip istemediği sorularak, 3 boyutlu bir yazıcıyla entegre edilerek yazdırılabilecek halde geliştirilebilir.

## 7. KAYNAKLAR

- [1] Demir H., Savaş V. ve Altuğ M., “Tersine Mühendislik Yöntemi ile Cad Datası Oluşturma: Femur Kemiği Örneği”, *Batman Üniversitesi Yaşam Bilimleri Dergisi* Cilt 9 Sayı 1, 2019
- [2] Balta C., “Tersine Mühendislik Uygulamalarında Nokta Bulutu Verilerinden Örne Yüzey ve Parametrik Yüzey Denklemlerinin Elde Edilmesi”, Doktora Tezi, Kocaeli Üniv., Kocaeli, 2016
- [3] Eragubi M., “Slicing 3D CAD Model in STL Format and Laser Path”, *International Journal of Innovation, Management and Technology*, 2013
- [4] Duman B. ve Kayacan M. C., “Eklemeli İmalatta Kullanılan STL Dosyalarının Hataları ve Onarım Yöntemleri”, *3Boyutlu Baskı Teknolojileri Sempozyumu (3D-BTS)*, İstanbul, 05-07 Mayıs 2016
- [5] Lueth T.C. and Irlinger F., "Berechnete Erzeugung von dreidimensionalen Oberflächenmodellen im STL-Format aus der Beschreibung planarer Mechanismen für die Generative Fertigung durch Selektives-Lasersintern [Computational 3D Surface Generation of Planar Mechanismus using STL File Format for Generative Manufacturing by Selective Laser Sintering]", *Angenommener Beitrag im Konferenzband 10. Kolloquium Getriebetechnik*, TU Ilmenau, 2013
- [6] Balachandar K., et al., “Macro Assisted Design Automation of Piston & Connecting Rod Using Solidworks API”, *TEST Engineering & Management*, ISSN: 0193-4120 Page No. 12964 – 12969, May-June 2020
- [7] Reverse engineering. n.d, [Online], Available: [https://en.wikipedia.org/wiki/Reverse\\_engineering](https://en.wikipedia.org/wiki/Reverse_engineering)
- [8] Raja, V., “Introduction to Reverse Engineering”. in *Raja, V. and Fernandes, K., Reverse Engineering. Springer Series in Advanced Manufacturing*. Springer, London. 2008, [https://doi.org/10.1007/978-1-84628-856-2\\_1](https://doi.org/10.1007/978-1-84628-856-2_1)
- [9] Türkücü T. ve Börklü H.R., “Tersine Mühendislik Yaklaşımına Dayalı Yeni Bir İmalat İçin Tasarım İşlem Modeli”, *GU J Sci, Part C*, 6(1): 91-104, 2018, <https://doi.org/10.29109/http-gujsc-gazi-edu-tr.327479>
- [10] Reverse Engineering Of Physical Objects – Teaching Manual, 2014. [Online]. Available: [https://www.creaform3d.com.cn/sites/default/files/assets/technological-fundamentals/teaching\\_manual\\_reverse\\_engineering\\_en\\_18032014\\_4.pdf](https://www.creaform3d.com.cn/sites/default/files/assets/technological-fundamentals/teaching_manual_reverse_engineering_en_18032014_4.pdf)

- [11] Mostafaei A., et al., “Binder jet 3D printing—Process parameters, materials, properties, modeling, and challenges”, in *Progress in Materials Science*, Mostafaei A., et al., Amsterdam: Elsevier, June 2021
- [12] Büyükkaraca T. N., Canan S. ve Baykan K. Ö., “3D dilimleme Yöntemi ile Kalıp Çıkarma”, *4th International Congress on 3d Printing (Additive Manufacturing) Technologies and Digital Industry*, Antalya, 11-14 Nisan 2019
- [13] Nelli F., “The STL file”, January 2014. [Online]. Available: <http://www.meccanismocomplesso.org/check-stl-files-3d-printing/>

# **EKLER**

## EKLER

### EK A: Matlab Kodları

```
clc;

clear;

fid=fopen('deliklicisim.STL','rt'); %çizdirilmek istenen STL dosyasının ismi girilmelidir
if fid<0
    error('error opening file %s\n\n',deliklicisim.STL);
end
i=1; j=1; k=3;

while feof(fid)==0
    L=fgets(fid);
    [L,count] = fscanf(fid, ['%*s %f %f %f', 3]);
    G{i}=L;
    i=i+1;
end

%vertex lerden 3x3 matrislerini oluşturma
for j=1:1:i
    if k<i
        M{j}=[G{k} G{k+1} G{k+2}];
    end

    % üçgenleri çizdirme
    plot3(M{j}(1:2,1),M{j}(1:2,2),M{j}(1:2,3),'r')
    hold all
    plot3(M{j}(2:3,1),M{j}(2:3,2),M{j}(2:3,3),'g')
    plot3(M{j}(1:2:3,1),M{j}(1:2:3,2),M{j}(1:2:3,3),'b')

    k=k+6;
end
end
fclose(fid);
```

## EK B: Solidworks Api/Makro Kodları

Option Explicit

```
Dim swApp As SldWorks.SldWorks 'swApp uygulamasını solidworks e tanımlama  
Dim swmodel As SldWorks.ModelDoc2 'modeldoc2 da tanımlama  
Dim boolvalue As Boolean
```

---

Sub silindir()

```
Set swApp = Application.SldWorks  
Set swmodel = swApp.ActiveDoc  
boolvalue = swmodel.Extension.SelectByID2("Üst Düzlem", "PLANE", 0, 0, 0, False, 0,  
Nothing, 0) ' Düzlem oluşturma
```

```
swmodel.SketchManager.InsertSketch True 'sketch oluşturma
```

```
Dim skSegment As Object
```

```
Set skSegment = swmodel.CreateCircleByRadius2(0, 0, 0, 0.05) 'silindir oluşturma 0.05 m
```

```
boolvalue = swmodel.Extension.SelectByID2("", "SKETCHSEGMENT", 0, 0, 0, False, 0,  
Nothing, 0) 'Katılaştırma için sketch seçimi
```

```
Dim myfeature As Object
```

```
Set myfeature = swmodel.FeatureManager.FeatureExtrusion2(True, False, False, 0, 0, 0.1,  
0.1, False, False, False, False, 1.74532925199433E-02, 1.74532925199433E-02, False,  
False, False, False, True, True, True, 0, 0, False) ' katılaştırma işlemi
```

```
End Sub
```



## ÖZGEÇMİŞ

### Kişisel Bilgiler

Adı Soyadı : Mehmet İNCE

Doğum tarihi ve yeri : **02.06.1983**

e-posta : mehmettince@hotmail.com

### Öğrenim Bilgileri

Derece	Okul/Program	Yıl
Y. Lisans	Balıkesir Üniversitesi/Elektrik Elektronik Mühendisliği	2023
Lisans	Zonguldak Bülent Ecevit Üniversitesi/ Elektrik Elektronik Mühendisliği	2007
Lise	Mersin Gazi Süper Lisesi	2001

### Mesleki Bilgiler

Şirket	Unvan	Yıl
Türk Telekomünikasyon A.Ş	Saha Operasyonları Ekip Lideri	2008-....
Erbim Elektrik Ltd. Şti	Koordinatör Mühendis	2007-2008