# **Programlama Laboratuvar Raporu**

**Minimum Çevreleyen Çember**

Berke YILDIZ  
Kocaeli Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği [190202049@kocaeli.edu.tr](mailto:180202025@kocaeli.edu.tr)

Çağrı ŞENTRÜK   
Kocaeli Üniversitesi  
Mühendislik Fakültesi Bilgisayar Mühendisliği [190202055@kocaeli.edu.tr](mailto:190202055@kocaeli.edu.tr)

**Özet- Programlama Laboratuvarı 1.Proje olarak bizden verilen koordinat noktalarını koordinat düzleminde gösterip, bu koordinat noktalarını kapsayan en küçük çemberi, bu çemberin merkez noktasını ve çemberin yarıçapını koordinat düzleminde çizdirmemizi ayrıca bu çemberin merkez noktasını ve yarıçapını ekrana bastırmamız isteniyor. Bu işlemlerden sonra koordinat noktalarının her birinin yakınından geçen eğriyi çizdirip kodumuzu tamamlamamız bekleniyor. Bu proje için C dilini ve Dev-C++ geliştirme ortamını seçtik. İlk olarak projede belirtildiği gibi .txt dosyasından verilen koordinatları dizi yardımıyla okuyup belleğe yazdık. Daha sonrasında bu noktaları belirteceğimiz koordinat düzlemi için graphics.h kütüphanesi yardımıyla bir koordinat düzlemi oluşturduk. Noktalarımızı oluşturduğumuz koordinat düzlemine yerleştirdik ve daha sonrasında matematiksel işlemler kullanarak bu noktalarımızı kapsayan en küçük çemberi saptadık. Bu çemberimizi de çizdirdikten sonra son olarak bu noktalarımızın yakınından geçen eğrimizi çizdirerek kodumuzu tamamladık.**

# **I.PROJE HAKKINDA**

Projemiz birkaç aşamadan oluşmaktadır ve her aşamada isterler bulunmaktadır . Proje kaba olarak 2 boyutlu düzlem için gerekli olan N adet noktanın koordinatlarını .txt dosya formatından okuma yaparak. Her bir X ve Y koordinatlarını ileriki işlemlerimiz için belleğe kaydettik. Kaydettiğimiz noktaları selection sort algoritması ile x değerlerine göre küçükten büyüğe sıraladık. Daha sonra bir arayüz aracılığı ile koordinat düzlemimizi ekrana çizdirdik. Bellekte tutulan (X,Y) koordinatlarını koordinat düzlemine uygun olarak yerleştirip ekranda gösterdik. Bu noktalarımızı kapsayan en küçük çemberimizin yarıçapını ve merkez noktasını matematiksel işlemler yardımı ile bulduk. Bu çemberimizi ekrana çizdirdik. Matematiksel işlemlere raporumuzun ilerisinde değinilecektir. Bulduğumuz yarıçap ve merkez noktasını konsola yazdırdık. Bizden tüm noktaların en yakınından geçen bir eğrinin çizdirilmesi istenmektedir. Bu eğriyi çizdirmek için B-spline yöntemine başvurduk. Bulduğumuz eğriyi koordinat düzlemimize çizdirdik. B-spline yönteminin matematiksel işlemlerine raporumuzun ilerisinde değinilecektir.

# **II.YÖNTEMLER VE PROGRAM MİMARİSİ**

## **A.Dosya İşlemleri**

N adet noktanın içinde bulunduğu txt dosyasını fopen() kullanarak açtık .Dosyamız yoksa dosya bulanamadı yazdırıp projeden çıkış yaptırdık. Kullanıcıdan girilen noktalarımızın adedini bilmediğimiz için dosyamız bitene kadar okuma işlemi yaptık. Bu işlemi döngü kullanarak dosyamızın sonuna gelene kadar koordinatlarımızı okuttuk. Dosyamızın sonuna geldiğinde ise dosyamızı kapattık.

## **B.Verilerin Okunması ve Saklanması**

Txt dosyasından okuduğumuz koordinatları int tipinde iki boyutlu dizi de sakladık. Bu sayede koordinatlarımız ile ilgili işlemlerimizde sürekli okuma yapmak yerine dizimizi kullanarak okuma zorunluluğumuzu ortadan kaldırdık. Bu sayede kodumuzu işlem fazlalığından kurtardık ayrıca koordinatlarımızı tek bir dizide tutmak ileriki işlemlerimizi karışıklık olmadan yapabilmemize olanak sağladı.

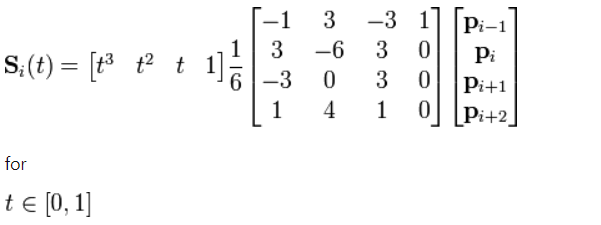
## **C.Matematiksel İşlemler**

**C.1-) En Küçük Çemberin Merkez Noktası ve Yarıçapı**

Txt dosyasından okuduğumuz koordinatları kullanarak istenilen çıktıları elde edebilmek için verilerimizi farklı matematiksel işlemlere sokmamız gerekiyordu. Noktaları kapsayan en küçük dairenin merkez noktası ve yarıçapını bulmak için matematiksel işlemleri oldukça fazla kullandık. İlk olarak girilen koordinatlarımızın X ve Y değerlerinin aritmetik ortalamasını aldık ve bunu ilk merkez noktamız olarak kabul ettik. Bunun sebebi en küçük çemberin merkez noktasını bulurken noktalarımıza çok uzak bir yerden başlamayıp daha az işlem yapmaktır.Daha sonra Pisagor Teoremi’ni merkez noktamız ile noktalar arası uzaklığı bulmak için kullandık. Bu uzaklıkların en büyüğününü merkezimize en uzak nokta olarak kabul ederek. Bu noktayı merkez noktamızın X ve Y sinin farkını alarak ulaştığımız bu sonuçları merkez koordinatımıza ekleriz. Eğer çıkan sonuç pozitif ise merkezimiz o noktaya yaklaşır. Sonuç negatif ise merkezimiz o noktadan uzaklaşır. Eklemeden önce döngümüzde sürekli küçülteceğimiz bir değişken ile çarparız. Çarpmamızın sebebi daha küçük sayılarla artırıp azaltarak daha hassas işlemi yaparak tam sonucu elde etmektir. Döngümüz bittiğinde ise en küçük çemberimizin merkezini elde ederiz. Merkezimizin en uzak nokta ile arasındaki uzaklık yarıçapımız olur.

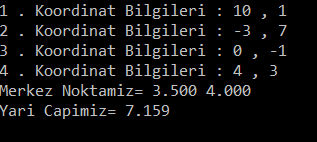
**C.2-) Kübik B-spline Yöntemi**

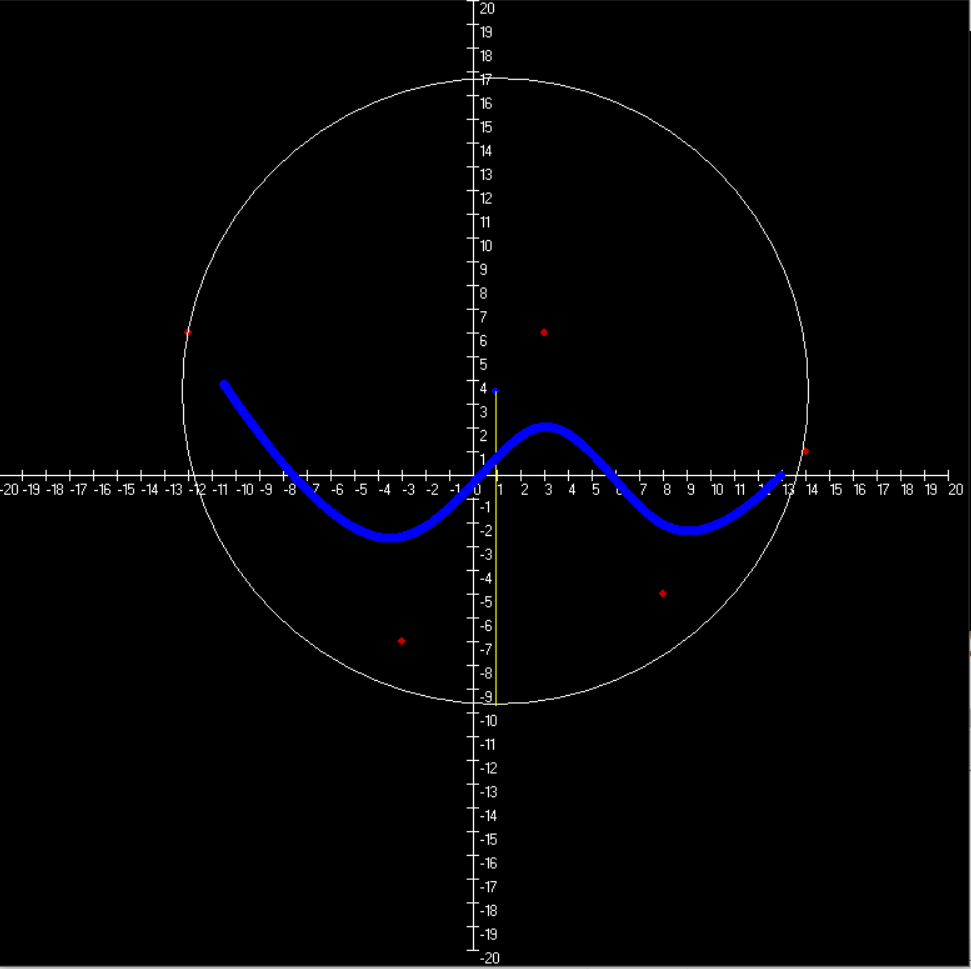
B-spline yöntemimizde ilk işlemi yapabilmek için 4 noktaya ihtiyacımız vardır ama kodumuz 4’ten daha az değer girilmesi durumunda da çalışabilmesi için ilk noktamızı 4 defa alırız. Daha sonra noktalarımızı aşağıda açıklayacağımız b-spline için gerekli matematiksel işlemleri yaptığımız fonksiyona yollarız.



Bu fonksiyonda yukarıdaki matris işlemini uygulamamız gereklidir. Fakat 3 matrisi aynı anda işleme tabi tutmak hem zor hem de karmaşık bir işleme yol açar. Bu nedenle ilk olarak 2. ve 3. matrisler arasında işlem yaptık, başındaki 1/6 katsayısından dolayı 6 ile bölme işlemi uyguladık ve bunları a ve b adlı dizilerde tuttuk. Bu işlemler sonucunda elimizde 1. matrisimiz ve a ile b dizilerinden oluşan yeni matrisimiz kaldı. Bu işlem için döngülerden yararlandık böylece işlem karmaşıklığından kurtulduk. Bu döngümüzü karşılaştırdığımız ardışık iki noktamız arasındaki uzaklık kadar devam ettirdik çünkü bu noktalarımız arasında eğri çizdirmemiz için bu uzaklık kadar işlem yapmamız gerekliydi. Bu işlemdeki t değerimiz yukarıdaki görselden anlaşılacağı gibi 0 ve 1 arasında bulunmalıydı. bu nedenle döngüdeki i değişkenimiz ile noktalarımız arasındaki uzaklığı bölerek 0 ve 1 arasında işlemler yaptık. Bunu yapmamızın sebebi en küçük noktaya kadar işlemlerimizi yapıp daha sonra bulduğumuz noktaları çizdirip eğrimizi doğru bir şekilde çizdirmektir. Daha sonra bu işlemleri tamamlayıp döngü yardımıyla aldığımız sonuçları ana fonksiyonumuza eğrimizi çizdirmek için yolladık.

**D.Program Çıktıları**

Dosyadan aldığımız verileri matematik işlemlerine sokarak bulduğumuz sonuçları ekrana çıktı olarak yazdırdık. Daha sonra bulduğumuz çemberimizin merkez noktasını ve yarıçapını buna ek olarak koordinat noktalarımızı göstermek için koordinat düzlemine ihtiyacımız vardı. Çemberimizi,noktalarımızı ve koordinat düzlemimiz graphics.h kütüphanesi ile çizdirdik. 

 Program Çıktılarımız

## **E.Programın Genel Yapısı ve Tasarım**

Programın genel yapısı ve işleyiş algoritması raporun en sonunda verilmiştir. Tasarım olarak standart bir konsol uygulaması biçimindedir.

## **F.Performans**

Progaramımızda çok sayıda döngü işlemi bulunmaktadır.Bunların birçoğu O(N) karmaşıklığa sahiptir ama koordinatlarımızı x’e göre küçükten büyüğe sıraladığımız iç içe döngümüz O(N²) zaman karmaşıklığına sahiptir.Zaman karmaşıklığının O(N²) olmasının sebebi döngümüzün toplamda N(N-1)/2 defa işlem yapmasıdır.

# **III.GELİŞTİRME ORTAMI VE KULLANILAN DİL**

Projeyi C programlama dilinde Windows işletim sistemi üzerinde gerçekleştirirken, geliştirme ortamı olarak cpp derleyicili Dev-C++ kullandık. Programda kullandığımız kütüphaneler

* Graphics.h
* Stdlib.h
* Math.h

# **IV.DENEYSEL SONUÇLAR**

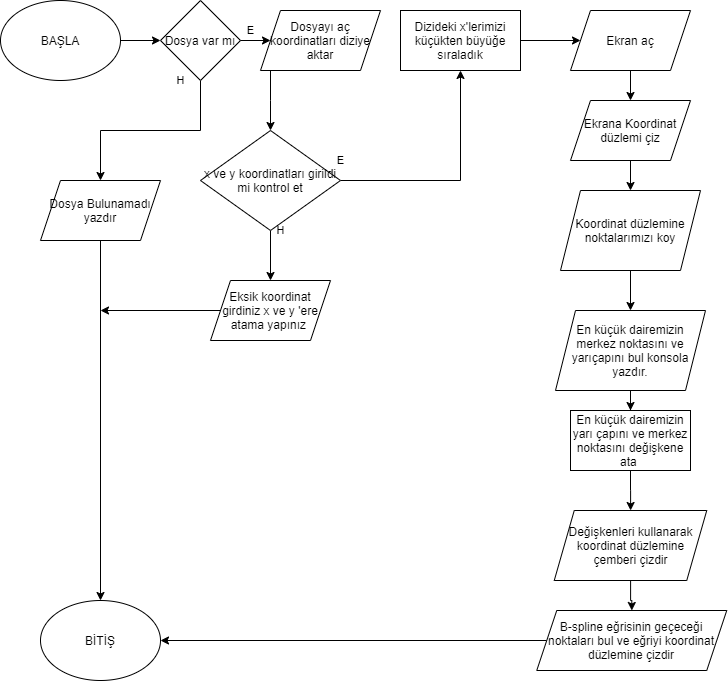
Çemberimizin merkez noktasını ve yarıçapını bulmak için ilk olarak çember analitiğini araştırmaya başladık fakat kodumuzda denediğimizde her zaman doğru sonuç almadığımızı gözlemledik. Araştırmaya ve denemeye devam ederek çemberimiz için en doğru algoritmayı bulduk.Bu algoritmayı koordinat düzlemimizde çizdirerek doğruluğunu gözlemlemiş olduk. B-spline eğrisini düzgün çizdirdip çizdiremediğimizi anlamak için koordinat sistemimizde çizdirerek doğruluğundan emin olduk.

# **V.SONUÇ**

Bu proje ile bir çok şey öğrendik. C dilinde graphics.h kütüphanesi kullanarak koordinat düzlemi çizdirmek ve bunu ölçeklendirebilmek hakkında bilgi sahibi olduk. Graphics.h kütüphanesinin fonksiyonlarını kullanmayı öğrendik ve bu sayede kodumuz grafiksel bir hale getirdik. Bu proje için yaptığımız araştırmalar sayesinde C dili hakkında yeni bilgiler öğrendik. C diline daha hakim olduk.Algoritma kurma ve çözümleme yeteneğimizi geliştirdik. Eğrimizi çizerken eğri çizme hakkında bilgi edindik ve mantığını kavradık.Eğrimizi çizdirirken kullandığımız matematiksel işlemler sayesinde Lineer Cebir konusu hakkındaki bilgilerimizi tazeledik

# **VI.KAYNAKLAR**

* <https://ckaynak.com/c-programlama-dili-winbgim-turkce-kaynak-grafik-ekrani>
* <https://fatihalparslann.wordpress.com/2017/02/02/c-dilinde-graphics-h-kutuphanesi-1-bolum/>
* <https://www.youtube.com/watch?v=WGOSX6d17TU>
* <http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm>
* <https://web.itu.edu.tr/yukselen/HM504/02Ek-%20Bezier%20e%F0rileri.pdf>
* Lineer Cebir ve Çözümlü Problemler Umuttepe Yayınları



Genel Akış Şeması