

## 1. Projenin Amacı

Bu çalışmanın amacı, kullanıcı tarafından girilen **4 adet kontrol noktasından geçen ve 3 parçalı** ( $1 \rightarrow 2$ ,  $2 \rightarrow 3$ ,  $3 \rightarrow 4$ ) süreklilik gösteren bir eğriyi, **tension (gerilim)** parametresi ile şekillendirerek ekranda çizdirmektir.

Kullanıcı x-y koordinatlarını ve tension değerini değiştirdiğinde **ReDraw** butonuna basarak eğriyi yeniden oluşturabilmektedir.

Bu proje; **HTML5 Canvas** ile görselleştirme, **JavaScript** ile arayüz/çizim kontrolü ve **VBScript** ile polinomsal hesaplamalar mantığını göstermektedir.

## 2. Kullanılan Teknolojiler ve Araçlar

- **Visual Studio:** Kod yazma ve düzenleme
- **BabyWeb.exe:** Classic ASP sayfayı localhost üzerinden çalıştırma
- **HTML5 Canvas:** Çizim alanı
- **JavaScript:** Arayüz (input okumaları, canvas çizimleri, ReDraw kontrolü)
- **VBScript (Classic ASP içinde):** Eğri noktalarının hesaplanması (server-side)

## 3. Uygulama Arayüzü

Arayüz iki ana bölümden oluşur:

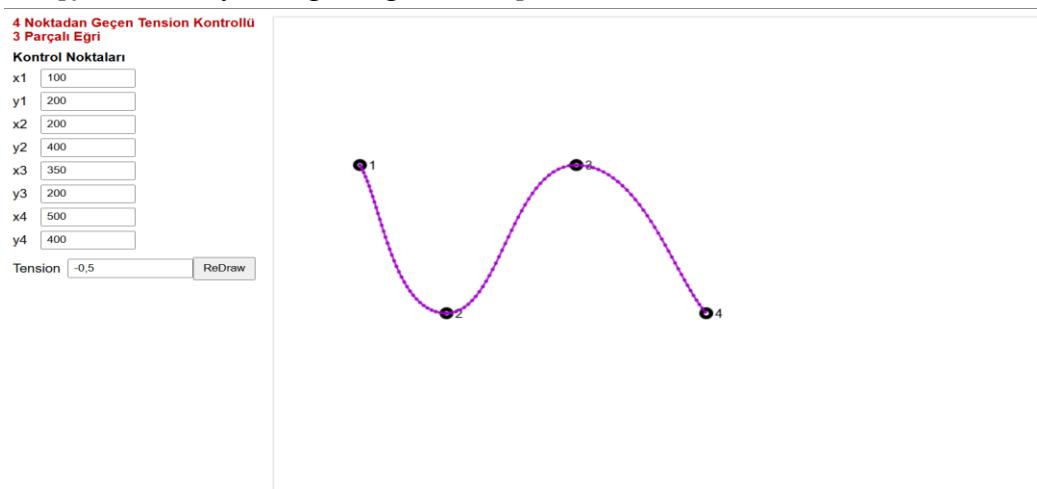
### 1. Kontrol Paneli

- $x1,y1 \dots x4,y4$  giriş kutuları
- tension giriş kutusu
- ReDraw butonu

### 2. Canvas Alanı

- Kontrol noktaları (1,2,3,4) numaralandırılmış şekilde çizilir
- Hesaplanan eğri noktaları çizilerek eğri oluşturulur

➡ [Şekil 1: Arayüzün genel görünümü]



## 4. Matematiksel Model ve Kullanılan Formüller

### 4.1. Tension Parametresi ve Teğet ( $rx$ , $ry$ ) Hesabı

Tension, eğrinin “gevşek / gergin” görünümünü etkiler. Öncelikle her nokta için teğet bileşenleri hesaplanır.

Ortak katsayı: **k=0.5(1-tension)**

Teğet hesapları:

- $rx(1) = k(x(2) - x(1))$   
 $ry(1) = k(y(2) - y(1))$
- $rx(2) = k(x(3) - x(1))$   
 $ry(2) = k(y(3) - y(1))$
- $rx(3) = k(x(4) - x(2))$   
 $ry(3) = k(y(4) - y(2))$
- $rx(4) = k(x(4) - x(3))$   
 $ry(4) = k(y(4) - y(3))$

Bu değerler her parçada Hermite polinomunda “eğimin” oluşmasını sağlar.

### 4.2. 3 Parçalı Kübik Hermite Eğrisi

Eğri 3 parçadan oluşur:

- Parça 1: Nokta1 → Nokta2 (i=1)
- Parça 2: Nokta2 → Nokta3 (i=2)
- Parça 3: Nokta3 → Nokta4 (i=3)

Her parça için  $t$  parametresi 0'dan 1'e gider:

- $t=0 \rightarrow$  parçanın başlangıç noktası
- $t=1 \rightarrow$  parçanın bitiş noktası

Adım: **step=0.03**

Hermite baz fonksiyonları:

$$h_{00}(t) = 2t^3 - 3t^2 + 1$$

$$h_{01}(t) = -2t^3 + 3t^2$$

$$h_{10}(t) = t^3 - 2t^2 + t$$

$$h_{11}(t) = t^3 - t^2$$

Her  $t$  için hesaplanan koordinatlar:

$$xx = x(i)h_{00} + x(i+1)h_{01} + rx(i)h_{10} + rx(i+1)h_{11}$$

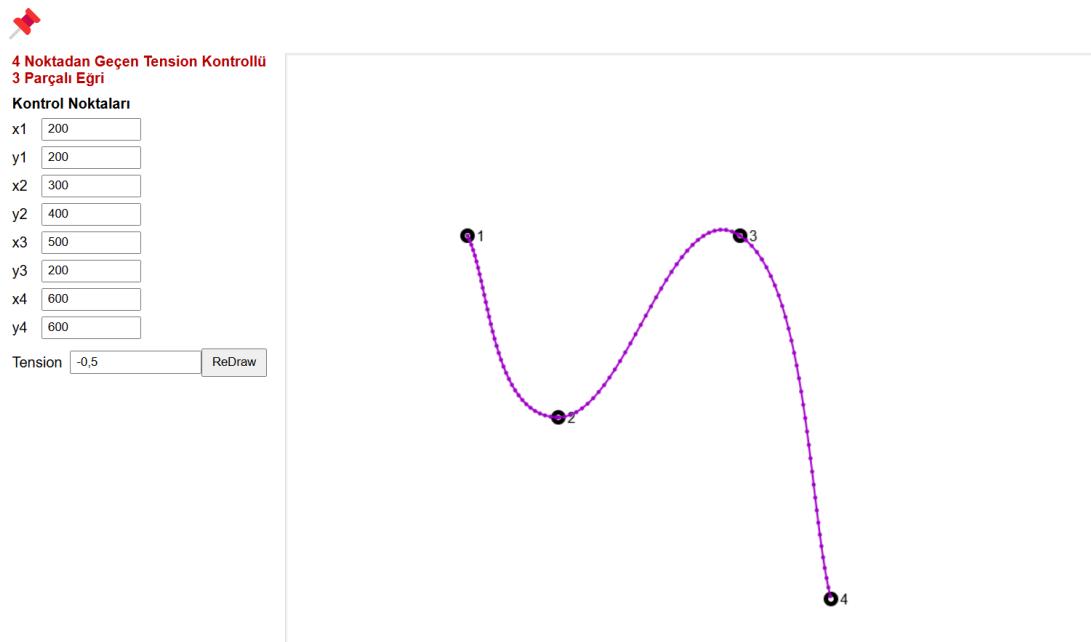
$$yy = y(i)h_{00} + y(i+1)h_{01} + ry(i)h_{10} + ry(i+1)h_{11}$$

Bulunan  $(xx, yy)$  noktaları canvas üzerinde çizilerek eğri elde edilir.

## 5. Programın Çalışma Mantığı

### 5.1. ReDraw Aksı

1. Kullanıcı koordinatları ve tension değerini girer.
2. **ReDraw** butonuna basılır.
3. JavaScript canvas'ı temizler ve kontrol noktalarını çizer.
4. JavaScript, sayfaya `?calc=1` parametresiyle istek atar.
5. Classic ASP tarafında VBScript çalışır:
  - a.  $rx, ry$  hesaplanır
  - b.  $i=1..3$  için  $t=0..1$  step 0.03 ile  $xx, yy$  üretilir
  - c. Sonuç JSON olarak döndürülür
6. JavaScript gelen noktaları alır:
  - a. noktaları birleştirerek çizgi halinde eğriyi çizer



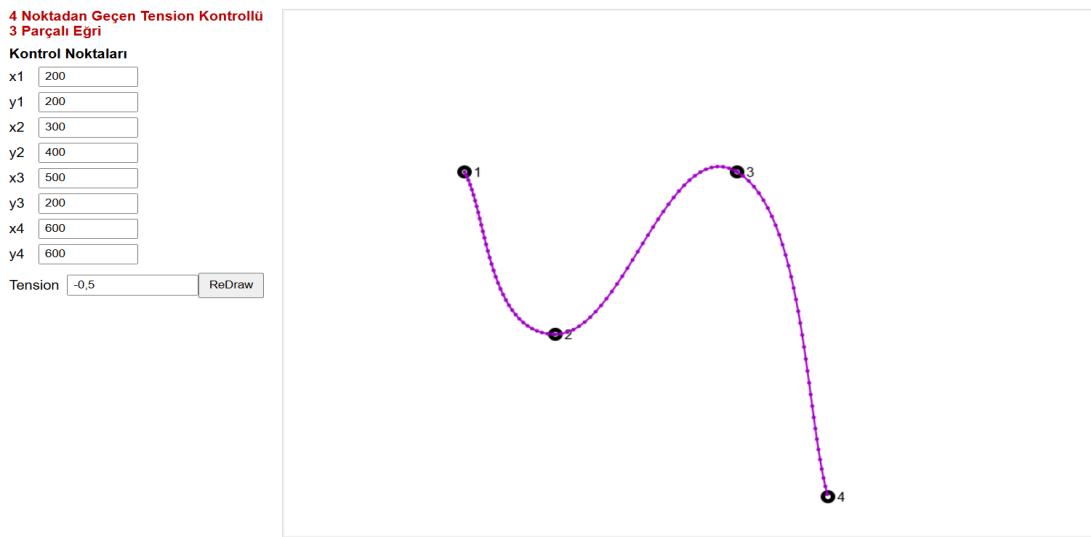
[Şekil 2: ReDraw sonrası eğri çizimi]

## 6. Tension Değerinin Etkisi (Gözlem)

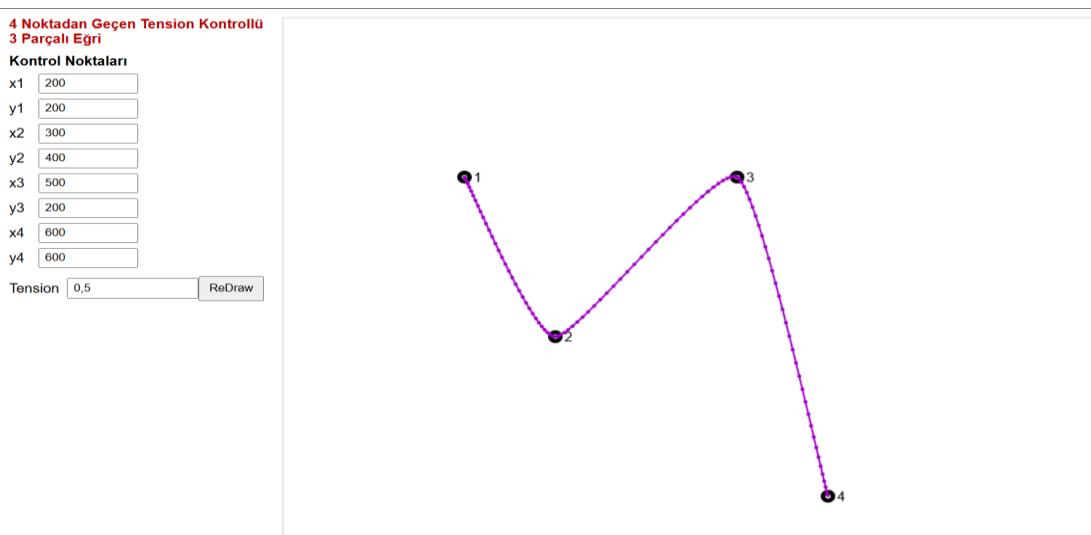
Farklı tension değerleri ile eğrinin davranışını gözlemlenmiştir:

- **tension negatif** (örn: -0.5): Eğri daha “yuvarlak / dalgalı” görünür.
- **tension 0 civarı**: Daha standart spline görünümü
- **tension pozitif** (örn: 0.5): Eğri daha “gergin / köşelere yakın” görünür.

### ❖ [Şekil 3: tension=-0.5 örneği]



### ❖ [Şekil 4: tension=0.5 örneği]



## 8. Sonuç

Bu projede 4 kontrol noktasından geçen, 3 parçalı kübik Hermite spline eğrisi tension parametresi ile kontrol edilerek başarıyla çizdirilmiştir. Kullanıcı, noktaları ve tension değerini değiştirip ReDraw ile dinamik olarak yeni eğri oluşturabilmektedir. Proje, polinom tabanlı eğri üretimi ve canvas görselleştirmeyi birlikte göstermektedir.

**2110205039 -Beyzanur Dulek**

**Bilgisayar Mühendisliği I. Öğr.**