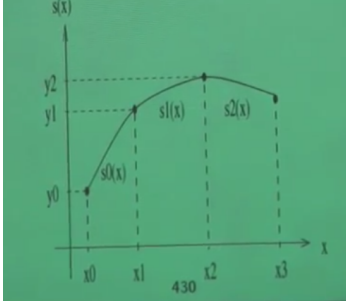


## Spline Egrileri

Diyelim ki elimizde 4  $x_i, y_i$  noktası var, ve bu noktalardan geçen, hepsinden *kesinlikle* geçen, yaklaşıksal bir eğri oluşturmak istiyoruz. Spline yöntemi her iki nokta arasını farklı bir küpsel (üçüncü derece) polinom ile temsil etmektir. Tekrar dikkat: tüm noktaları temsile edebilecek farklı polinomları toplamıyoruz, her aralıkta başka bir polinom fonksiyonu parçasını devreye sokuyoruz. Parçalar niye küpsel olarak seçildi? Çünkü küpsel bir eğri yeterince kavis sağlayabilir ve aynı zamanda çok fazla inisli çıkışlı, sivri değildir, yeterince pürüzsüz bir eğrinin ortaya çıkmasını sağlar.



Her  $i = 0, \dots, n + 1$  için

$$p(x) = p_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (1)$$

kullanalım. Noktalar  $x_i$  olarak gösteriliyor, ve her noktada aktif olan bir  $p_i$  spline olacak, o noktadan bir sonrakine kadar eğriyi bu  $p_i$  tanımlayacak. Noktaların sayısını  $n$  yerine  $n + 1$  olarak aldık böylece  $n$  eğri parçası ile çalışmamız mümkün olacak. Her spline bir kübik polinom ise niye bu kübik polinomu en basit şekliyle

$$p(x) = a_i + b_i x + c_i x^2 + d_i x^3$$

olarak tanımlamadık? Çünkü iki üstteki form ile çalışmak daha rahat. Mesela, eğer  $x$  için  $x_i$  değeri verirse, ki bu  $x_1$  ya da  $x_2$  olabilir, o zaman parantez içinde  $x_i - x_i$  sayesinde tüm terimler sıfır oluyor, geriye sadece  $a_i$  kalıyor.

Parçaların uçlarının birbirini tutması, ve tüm şeklin sürekli, akışkan bir şekilde gözükmesi için ise birkaç koşulu bizim tanımlamamız, ve zorlamamız gerekli. Önce en basit olanı: bir önceki parça ile bir sonraki parça orta nokta üzerinde aynı değere sahip olmalı.  $i = 1, \dots, n + 1$  için

$$p_i(x_{i+1}) = p_{i+1}(x_{i+1})$$

Bir diğer basit gereklilik, her  $x_i$ 'ye kabul eden spline fonksiyonun elimizdeki  $y_i$  değerini vermesi,

$$p_i(x_i) = y_i$$

“Tum noktalardan kesinlikle gecmeli” demistik. Son parca bir istisna olusturuyor, bu son parcanin fonksiyonu hem son noktayi, hem de ondan bir onceki nokta icin kullanilmali, bir onceden en sona kadar ayni fonksiyon uzerindeyiz.

$$p_n(x_n) = y_{n+1}$$

Sistemi daha detayli olarak gormek gerekirse, tum denklemleri yazalim,

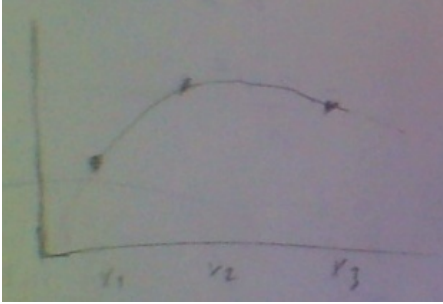
$$p_1(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3$$

$$p_2(x) = a_2 + b_2(x - x_2) + c_2(x - x_2)^2 + d_2(x - x_2)^3$$

⋮

$$p_n(x) = a_n + b_n(x - x_n) + c_n(x - x_n)^2 + d_n(x - x_n)^3$$

Uc noktali soyle bir grafik dusunelim,



Ustte bahsettigimiz gibi,  $p_1(x_1) = a_1 = y_1$  olacak, ve tum indisler icin bu gecerli. Ayrica  $x_2$  noktasinda bir onceki parca ve sonraki parca ayni degere sahip olmalı demistik, yani mesela  $p_1$ 'in sonunda (ustteki ilk parca)  $x_2$  noktası vardır, ve ayni noktada  $p_2$  baslayacaktır, o noktada

$$p_1(x_2) = a_1 + b_1h_1 + c_1h_1^2 + d_1h_1^3$$

ve bu denklem  $p_2(x_2) = a_2 = y_2$ 'ye esit. Bir de, daha once gorduk,  $a_1 = y_1$  ise, o zaman

$$y_2 = p_1(x_2) = y_1 + b_1h_1 + c_1h_1^2 + d_1h_1^3$$

haline gelir. Hepsini birarada yaziyoruz ( $y$ 'yi sag tarafa aldik)

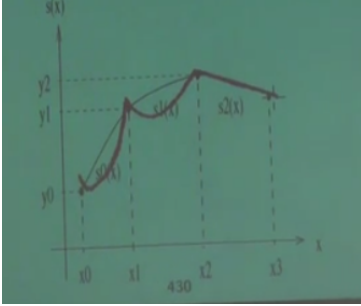
$$y_1 + b_1h_1 + c_1h_1^2 + d_1h_1^3 = y_2 \quad (2)$$

$$y_2 + b_2 h_2 + c_2 h_2^2 + d_2 h_2^3 = y_3$$

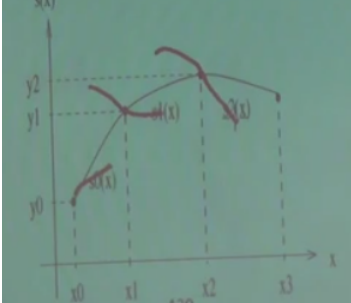
$\vdots$

$$y_n + b_n h_n + c_n h_n^2 + d_n h_n^3 = y_n$$

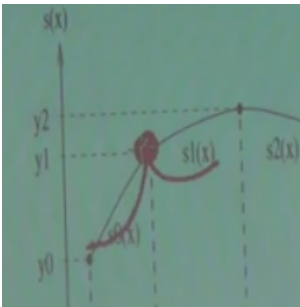
ki  $h_1 \equiv x_2 - x_1$ ,  $h_2 \equiv x_3 - x_2$  olarak tanımladık,  $\equiv$  isareti “tanımlamak (defined as)” anlamına geliyor,  $h$  harfi bir tur kısaltma olarak kullanıldı. Fakat kesintisizlik için parçaların uçlarının bitmesi yeterli değil. Mesela alttaki figürün de uçları birleşiktir,



Demek ki ek bazı şartlar lazım. Bu ek şart “sureklilik” olabilir. Mesela alttaki örnek sürekli değildir.

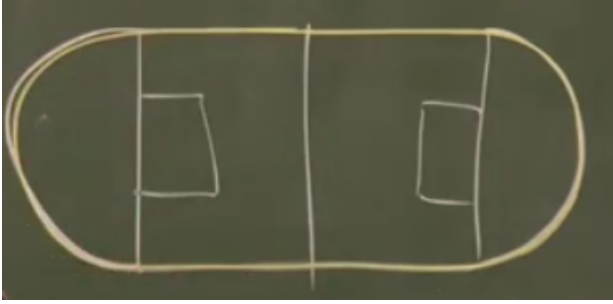


Ya da daha iyisi, fonksiyonun her noktada “turevi alınabilir” olma şartı. Mesela altta koyu yuvarlaklı gösterilen noktada fonksiyonun turevi alınamaz.

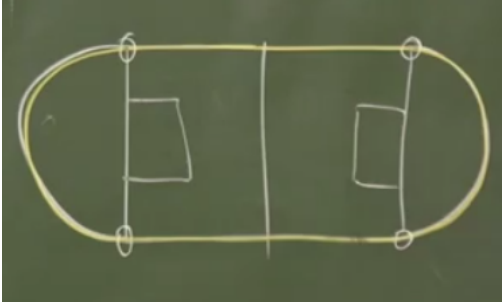


O zaman şartı koyalım – Fonksiyonun her noktasında, ikinci turev sürekli alınabilmeli. Bu çok ağır / net bir şart aslında, ve hakikaten çok pürüzsüz (smooth)

fonksiyonlara sebebiyet veriyor. Simdi bunun ne anlamina biraz daha yakindan bakalim. Bilirsiniz futbol sahalarinin etrafinda kosu alani vardir. Bu alan soyledir.



Bu sekil iki ayri figurun birlesimidir aslinda, duz cizgiler ve iki tane yari cember. Ustteki duz cizgili kisim sonsuz kere turevi alinabilir bir fonksiyondur. Degil mi? Duz cizgi sabit bir sayidir, 1. turev sifir, ikinci turev yine sifir, boyle gider. Peki yari cember olan kisimler? Ayni sekilde. Peki her noktada durum boyle midir? Kritik noktalar ufak yuvarlaklarla gosterilen yerler (altta)



Bu noktalarda kac kere “surekli turevler” alinabilir? Cevap, sadece bir kere. Cunku iki kere turev alinince ne olacagina bakalim, duz kisimda ikinci, ucuncu, vs. turev sifir. Peki yari cember? Onun ikinci turevi sifir olmayan sabit bir sayi. O zaman fonksiyonun tamaminin (duz cizgi ve yari cemberin beraber) 2. turevini grafikleseik, soyle bir sekil ortaya cikardi,



ve bu grafikte goruyoruz ki bir ziplama var. Bu ziplama yuzunden sureklilik (2. turevde) bozulmus oldu. O zaman spline duzgün, puruzsuz olsun istiyorsak, her noktada, yani baglanti noktalarinda, sagdaki ve soldaki parcanin birinci ve ikinci turevinin ayni olmasi sartini koyabiliriz, o zaman bu noktalarda fonksiyonun tamamı iki kere surekli turevi alinabilir hale gelir. Parcalarin kendisi uzerinde bu sarti tanimlamaya gerek yok, cunku orada polinom kullanacagimizi belirttik zaten, polinomlar

sonsuz kere surekli turevi alinabilen objelerdir.

Denklemlerimize iki tane daha sart gerekiyor. Bu sartlar fonksiyonun ilk noktada ve son noktada ikinci turevinin sifir olmasi sarti olabilir. Her hangi yondeki bir cizgi  $y = ax + b$ 'nin iki kere turevi alinince sifir gelir, yani bu sart fonksiyonumuzun son noktalarda, fonksiyonun “asagi yukari ayni yonde” olacak sekilde duz olarak devam etmesi anlamina geliyor. Yaklasiksal baglamda fena bir sart degil.

O zaman ana formullerimize donelim, ve mesela  $p_1(x), p_2(x)$ 'in turevini alalim,

$$p_1'(x) = b_1 + 2c_1h_1 + 3d_1h_1^2$$

$$p_2'(x) = b_2 + 2c_2h_2 + 3d_2h_2^2$$

$\vdots$

Turevleri esitleyelim  $p_1'(x_2) = p_2'(x_2)$ .

$$p_1'(x_2) = b_1 + 2c_1h_1 + 3d_1h_1^2$$

$$p_2'(x_2) = b_2$$

Ustteki niye sadece  $b_2$  oldu? Cunku  $x_i - x_i$  numarasi onun icin de gecerli, geriye sadece  $b_2$  kaldi. Hepsi bir arada

$$b_1 + 2c_1h_1 + 3d_1h_1^2 = b_2 \quad (3)$$

$$b_2 + 2c_2h_2 + 3d_2h_2^2 = b_3$$

$\vdots$

$$b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2 = b_n$$

Ikinci turevler icin benzer bir durum var, bu sefer sol taraftan  $b$ 'ler yokoluyor,

$$2c_1 + 6d_1h_1 = 2c_2 \quad (4)$$

$$2c_2 + 6d_2h_2 = 2c_3$$

$\vdots$

$$2c_{n-1} + 6d_{n-1}h_{n-1} = 2c_n$$

İlk ve son ikinci turevi sifira esitlemeyi unutmayalim. Son turev

$$2c_n + 6d_nh_n = 2c_{n+1} = 0$$

İlk turev

$$p_1''(x_1) = c_1 + 6d_1(x_1 - x_1) = c_1 = 0$$

$$6d_1(x_1 - x_1)$$

sifir olur

Denklem (4)'den baslayan blogu tekrar duzenlersek,

$$d_1 = \frac{c_2 - c_1}{3h_1} \quad (5)$$

$$d_2 = \frac{c_3 - c_2}{3h_2}$$

$\vdots$

$$d_n = \frac{c_{n+1} - c_n}{3h_n}$$

Ustteki denklemleri (2) ve (3)'e geri koyarsak,

$$b_1 + \frac{c_2 + 2c_1}{3}h_1 = s_1 \quad (7)$$

$$b_2 + \frac{c_3 + 2c_2}{3}h_1 = s_2$$

$\vdots$

$$b_n + \frac{c_{n+1} + 2c_n}{3}h_n = s_n$$

ki  $s_1 \equiv \frac{y_2 - y_1}{h_1}$ ,  $s_2 \equiv \frac{y_3 - y_2}{h_2}$ .

(3) ifadesini alip tekrar duzenlersek,

$$2c_1h_1 + 3d_1h_1^2 = b_2 - b_1$$

$3d_1h_1$  icin baska bir ifade kullanabiliriz, eger (5)'i tekrar duzenlersek,

$$3h_1d_1 = c_2 - c_1$$

ve iki ustteki formule koyarsak

$$2c_1h_1 + (c_2 - c_1)h_1 = b_2 - b_1$$

$$2c_1h_1 + c_2h_1 - c_1h_1 = b_2 - b_1$$

$$c_1h_1 + c_2h_1 = b_2 - b_1$$

$$(c_1 + c_2)h_1 = b_2 - b_1$$

Bu ifade tum  $i$  noktaları icin gecerli, hepsi bir arada

$$(c_1 + c_2)h_1 = b_2 - b_1 \quad (6)$$

$$(c_2 + c_3)h_2 = b_3 - b_2$$

$\vdots$

$$(c_{n-1} + c_n)h_{n-1} = b_n - b_{n-1}$$

(7)'deki ardi ardina gelen denklemleri birbirinden cikartip sonucu 3 ile carparsak,

$$c_1h_1 + 2c_2(h_1 + h_2) + c_3h_2 = 3(s_2 - s_1)$$

$$c_2h_2 + 2c_3(h_2 + h_3) + c_4h_3 = 3(s_3 - s_2)$$

$\vdots$

$$c_{n-1}h_{n-1} + 2c_n(h_{n-1} + h_n) + c_{n+1}h_n = 3(s_n - s_{n-1})$$

Bu formuller birarada dusunulurse, bilinmeyenleri  $c_2, c_3, \dots, c_n$  olan normal (ordinary)  $n - 1$  tane lineer denklemdirler, ve bir matris carpimi olarak dusunulebilirler.

$c_1 h_1$  matris formunda yok cunku  $c_1 = 0$ .

$$\begin{bmatrix} 2(h_1 + h_2) & h_2 & 0 & 0 & \dots & 0 \\ h_2 & 2(h_2 + h_3) & h_3 & 0 & \dots & 0 \\ 0 & h_3 & 2(h_3 + h_4) & h_4 & \dots & 0 \\ 0 & 0 & h_4 & 2(h_4 + h_5) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & h_{n-1} & 2(h_{n-1} + h_n) \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}$$

Bu denklem sag tarafta suna esit

$$\begin{bmatrix} 3(s_2 - s_1) \\ 3(s_3 - s_2) \\ 3(s_4 - s_3) \\ \vdots \\ 3(s_n - s_{n-1}) \end{bmatrix}$$

Bir ucgen kosegen (tridiagonal) matris iki tane ikili kosegen (bidiagonal) matrisin carpimina esittir. LU carpanlarına ayirma islemi de (bkz Lineer Cebir Ders 4) bize bu matrisleri saglayacaktır.

$$Ax = b$$

su hale gelir

$$LUx = b$$

Simdi eger  $Ux = y$  kabul edersek, yani yeni bir degiskeni dahil edersek,  $L$ 'i bulduktan sonra

$$Ly = b$$

kabul edebiliriz, ve bu formulu de  $y$  icin cozmek cok kolaydir. Sonra cozulen  $y$ 'yi alip geriye sokma (backsubstitution) ile  $x$ 'i buluruz, yani

$$Ux = y$$

denklemini cozeriz.



```

import scipy.linalg as lin

a = np.array( [[3.,-3.,0,0],
               [-3.,8.,-2.,0],
               [0,1.,2.,4.],
               [0,0,-2.,6.]] )

p,l,u = lin.lu(a)

Ly = np.array([[7.,8.,2.,-3.]])

y = lin.solve(l,Ly.T)

x = lin.solve(u,y)
print x

[[ 5.44047619]
 [ 3.10714286]
 [ 0.26785714]
 [-0.41071429]]

```

Spline yontemine donersek, elimizdeki veri ve kod soyle olsun

```

import scipy.linalg as lin

xx = np.array([4.,9.,12.,16.,22.])

yy = np.array([157.,41.,145.,92.,7.])

h = np.diff(xx)

dy = np.diff(yy)

s = dy / h

ds = np.diff(s)

s3 = 3 * ds

a = np.array([[ 2*(h[0]+h[1]), h[1], 0],
               [ h[1], 2*(h[1]+h[2]), h[2]],
               [ 0, h[2], 2*(h[2]+h[3])]])

p,l,u = lin.lu(a)

y = lin.solve(l,s3.T)

```

```
c = lin.solve(u,y)
print c
```

```
[ 13.45756677 -13.90702275   2.64390455]
```

$c$ 'ler bulunduktan sonra  $h$ 'lerle beraber kullanılarak  $d$ 'ler bulunur, vs, ve tüm spline parçalarının katsayıları ortaya çıkarılır.

Kaynaklar

<http://spartan.ac.brocku.ca/~jvrbik/MATH2P20/notes.pdf>

<http://www.youtube.com/watch?v=3rHBCg1D1LQ>

<http://www.youtube.com/watch?v=nA0YpgraP9A>