

# FÜGGVÉNYKÖZELÍTÉS SZEMLÉLTETÉSE AR-KÓDDAL

szakdolgozat

Bajcsay Kristóf Antal

Matematika BSc, alkalmazott matematikus szakirány

Témavezető: Dr. László Lajos, egyetemi docens

IK Numerikus analízis tanszék



Eötvös Lóránd Tudományegyetem  
Természettudományi Kar

Budapest, 2012

# Tartalom

Bevezetés .....	3
1. Interpoláció .....	4
1.1 Lagrange – féle interpolációs polinom.....	5
1.2 Newton – interpoláció .....	7
1.3 Hermite – interpoláció.....	9
1.3.1 Kétpontos Hermite – interpoláció.....	11
2 Spline – interpoláció .....	13
2.1 Természetes köbös spline – interpoláció.....	15
2.2 Másodfokú - spline.....	19
3 A program .....	21
3.1 Augmented Reality ( kiterjesztett valóság ) .....	21
3.2 A program részletes ismertetése.....	24
Összefoglalás .....	29
Melléklet .....	30
Irodalomjegyzék .....	34

## Bevezetés

Az életben számtalan olyan helyzet adódik, amikor adatok egy halmazából akarunk következtetéseket levonni, akár a jövőre akár csak egy két ismeretlen pontra vonatkozóan. Ezen problémakör gyakorlati megoldása jóval megelőzte a matematikai módszerek definiálását, a függvényközelítéseket.

Például ha méréseket kell végeznünk egy területen, de a környezet viszontagságai, vagy akár az idő hiánya miatt nem tudjuk ezt minden pontban megtenni. Ekkor elegendő néhány pontra elvégezni a mérést és az interpolációs eljárást. Az eredményül kapott függvény segítségével könnyedén meghatározhatjuk a keresett pontokban is a mérési értékeket.

A függvényközelítéseknek két csoportját különböztetjük meg. Az egyik az approximáció, a másik az interpoláció. Legyen  $X$  egy metrikus tér a  $d$  metrikával. Az approximációs eljárásához szükségünk van egy  $U \subset X$  halmazra, melyet az approximációs függvények halmazának tekintünk. Ebben az esetben minden  $f \in X$  -hez az eljárás talál egy  $u_f \in U$  -t, amit az  $f$  approximációjának tekintünk. A legjobb approximáció az az  $u^* \in U$ , melyre teljesül:

$$d(f, u^*) = \inf \{d(f, u) : u \in U\} = \text{dist}(f, U)$$

Ilyen  $u^*$  meghatározása gyakran nagyon időigényes, ezért helyette, olyan  $u_f \in U$  -t választunk, hogy  $d(f, u_f)$  közel legyen  $\text{dist}(f, U)$ -hoz.

Az interpolációnál a függvény nem ismert értékeire szeretnénk az ismert értékek alapján közelítést adni.

Szakedolgozatomban is ilyen interpolációs eljárásokat fogok bemutatni.

Először az egyszerűbb nem feltétlenül jó közelítéseket adó Lagrange, Newton interpolációs módszert ismertetem. Ezek után az Hermite interpolációt, ami a legelterjedtebb interpolációs eljárások alapjául is szolgál. Végül az előbbire épülő spline interpolációs eljárásokat. Ezek közül bővebben a természetes köbös spline interpolációt mutatom be.

Az interpolációs eljárások könnyebb megértése végett készítettem egy programot, mely a kiterjesztett valóság segítségével látványosabbá teszi ezek ismertetését. Dolgozatom harmadik részében erről a programról és az általa használt AR (augmented reality, magyarul kiterjesztett valóság) kódokról, ezek hasznáról lesz szó.

# 1. Interpoláció

Amikor interpolációról beszélünk, tulajdonképpen egy olyan feladat, illetve problémakörre kell gondolni, ahol célunk egy ismert, vagy ismeretlen függvény adott pontjai segítségével egy közelítést adni. A függvény közelítő polinomot hívjuk interpolációs polinomnak. Egy  $n+1$  pontból álló halmazra illeszthető egy  $n$ -edfokú polinom. Ennek bizonyításához legyenek adottak az  $x_0, x_1, x_2, x_3, \dots, x_n$  alappontok abszcisszái, melyek különbözőek és az  $f_0, f_1, f_2, f_3, \dots, f_n$  hozzájuk tartozó függvényértékek.

Megmutatjuk, hogy egyértelműen létezik hozzájuk interpolációs polinom.

Tegyük fel, hogy  $p(x)$  a keresett polinom, azaz:

$$p(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_n \cdot x^n.$$

Behelyettesítve az adott pontokat kapjuk az alábbi egyenletrendszert:

$$a_0 + a_1 \cdot x_k + a_2 \cdot x_k^2 + \dots + a_n \cdot x_k^n = f_k, \quad \text{ahol } k = 0, 1, 2, \dots, n$$

Az így kapott egyenletrendszer mátrixalakja a következő lesz:

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix}}_M \cdot \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}}_a = \underbrace{\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}}_f$$

Egyszerűbben írva:

$$M \cdot a = f$$

$M$  egy úgynevezett Vandermonde – mátrix melyről tudjuk, hogy reguláris, ami azt jelenti, hogy invertálható. Ebből pedig következik, hogy egyértelműen létezik a fenti egyenletrendszernek megoldása és a következő módon számítható ki:

$$a = M^{-1} \cdot f$$

Ezzel be is láttuk az állításunkat.

Azt mondjuk egy polinomra, hogy interpolációs, ha teljesíti az interpolációs feltételeket, vagyis:

$$p(x_k) = f_k, \quad k = 0, 1, 2, \dots, n$$

Az általános, sztenderd interpolációnál felmerül a probléma, hogy minél több pontot adunk meg a kezdeti feltételek között, annál magasabb fokú lesz az interpolációs polinom. A pontok közti távolság változtatásával, vagyis a megadott pontok tologatásával lehet a közelítésen javítani, de sajnos sok esetben inkább rontunk az eredményen, ugyanis ahogy azt a későbbi 1.ábra is mutatja egyetlen pont  $x$  koordinátájának megváltoztatásával a görbe a pontok között nagyon hullámossá válhat.

A továbbiakban néhány interpolációs eljárás és ezek eredményeként kapott interpolációs polinom bemutatásáról lesz szó.

## 1.1 Lagrange – féle interpolációs polinom

Először definiáljuk a Lagrange – interpolációs módszer alappilléret adó Lagrange – alappolinomokat.

$$l_k(x) := \prod_{\substack{j=0 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j}$$

Ezek könnyű, ámbar hosszadalmas számolással meghatározhatók.

A fenti képleten látható, hogy a számláló egy  $n$ -edfokú polinom, a nevező pedig egy konstans, vagyis  $l_k(x)$  pontosan  $n$ -edfokú.

Állítjuk, hogy az

$$L_n(x) := \sum_{k=0}^n f_k \cdot l_k(x)$$

képlettel megadott függvény interpolációs polinom, azaz teljesül rá, hogy  $L_n(x_i) = f_i$ . Az állítást egyszerűen bizonyíthatjuk, ugyanis az alappolinomok képletébe az alappontokat kapjuk, hogy:

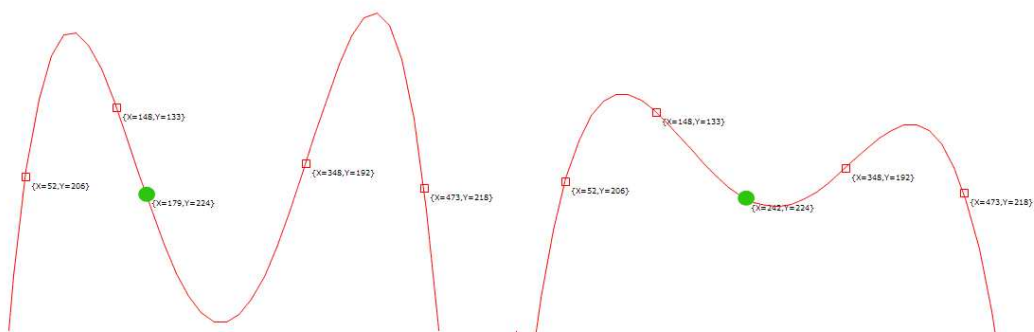
$$l_k(x_j) := \begin{cases} 1 & , \text{ ha } j = k \\ 0 & , \text{ ha } j \neq k \end{cases}$$

Ebből és  $L_n(x)$  definíciójából kapjuk, hogy tetszőleges  $i$ -re ( $i \in \{0,1,2,\dots,n\}$ )

$$L_n(x_i) = \sum_{k=0}^n f_k \cdot l_k(x_i) = f_i \cdot 1 = f_i$$

Vagyis az  $L_n(x)$  polinom tényleg interpolációs polinom.

Az alábbi ábra mutatja, hogy hiába illeszkedik jól az adott alappontokra a polinom, egy minimális változtatás is az alappontokban nagy hatással van a görbére.



1.ábra: Egyetlen alappont változtatásának hatása az illesztett polinomra

## 1.2 Newton – interpoláció

Ebben az esetben az interpolációs polinomot a következő alakban keressük:

$$L_n(x) = c_0 + c_1 \cdot (x - x_0) + c_2 \cdot (x - x_0) \cdot (x - x_1) + \dots + c_n \cdot (x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_{n-1})$$

ahol  $x_0, x_1, x_2, \dots, x_n$  a megadott alappontok abszcisszái. Ekkor mivel az alappontokhoz tartozó  $f_i$  függvényértékek is adottak  $L_n(x)$ -be az alappontok behelyettesítésével kapjuk:

$$L_n(x_0) = f_0 \quad \Rightarrow \quad f_0 = c_0$$

$$L_n(x_1) = f_1 \quad \Rightarrow \quad f_1 = f_0 + c_1 \cdot (x_1 - x_0), \quad \text{ahonnan} \quad c_1 = \frac{f_1 - f_0}{x_1 - x_0}$$

A fenti hányadost nevezzük elsőrendű osztott differenciának és  $f[x_0, x_1]$ -gyel jelöljük.

E jelölést felhasználva írható:

$$c_1 = f[x_0, x_1].$$

Tovább folytatva az alappontok behelyettesítését

$$L_n(x_2) = f_2 \quad \Rightarrow \quad f_2 = f_0 + c_1 \cdot (x_2 - x_0) + c_2 \cdot (x_2 - x_0) \cdot (x_2 - x_1), \quad \text{ahonnan}$$

$$c_2 = \frac{f_2 - f_0 - c_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$$

Az újonnan bevezetett jelölést használva  $c_2$  a következőképpen írható fel egyszerűbben:

$$c_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_1}$$

A jobb oldalon álló hányadost másodrendű osztott differenciának nevezzük és  $f[x_0, x_1, x_2]$  -vel jelöljük.

Alkalmazva ezt a jelölést  $c_2 = f[x_0, x_1, x_2]$ .

A  $k$ -adrendű osztott differenciát is hasonló számolások útján kaphatjuk meg.

Az együtthatók behelyettesítésével a következőt kapjuk  $L_n(x)$  - re:

$$L_n(x) = f_0 + f[x_0, x_1] \cdot (x - x_0) + f[x_0, x_1, x_2] \cdot (x - x_0) \cdot (x - x_1) + \dots \\ + f[x_0, x_1, \dots, x_n] \cdot (x - x_0) \cdot \dots \cdot (x - x_{n-1})$$

$L_n(x)$  meghatározásához a következő rekurzív eljárást használjuk:

$$L_0(x) = f_0$$

$$L_n(x) = L_{n-1}(x) + f[x_0, \dots, x_n] \cdot (x - x_0) \cdot \dots \cdot (x - x_{n-1}).$$

Egy példán keresztül szemléltetjük, hogy mennyire egyszerű ez a módszer.

Legyenek az alappontok rendre: -1 , 0 , 1 , 2 a hozzájuk tartozó függvényértékek pedig: -7 , -1 , 3 , 17 . Készítünk egy táblázatot az alábbi módon:

$x$	$f(x)$	differenciák	
$x_0 = -1$	$f_0 = -7$		
		$f[x_0, x_1] = \frac{6}{1}$	
$x_1 = 0$	$f_1 = -1$		$f[x_0, x_1, x_2] = \frac{-2}{2}$
		$f[x_1, x_2] = \frac{4}{1}$	$f[x_0, x_1, x_2, x_3] = \frac{6}{3}$
$x_2 = 1$	$f_2 = 3$		$f[x_1, x_2, x_3] = \frac{10}{2}$
.		$f[x_2, x_3] = \frac{14}{1}$	
$x_3 = 2$	$f_3 = 17$		



Ebben a táblázatban már minden megvan ahhoz, hogy  $L_3(x)$ -et kiszámoljuk.

$$L_3(x) = -7 + 6 \cdot (x+1) - 1 \cdot (x+1) \cdot (x-0) + 2 \cdot (x+1) \cdot (x-0) \cdot (x-1) = -1 + 3 \cdot x - x^2 + 2 \cdot x^3$$

Láthattuk tehát, hogy a Lagrange és a Newton interpoláció jó, bár sok alappont esetén a pontok között nagyon hullámos lesz a függvény, minek következtében a gyakorlatban nem lehet akármikor használni.

### 1.3 Hermite – interpoláció

Adottak  $x_0, x_1, x_2, x_3, \dots, x_n$  alappontok abszcisszái és  $m_0, m_1, m_2, m_3, \dots, m_n \geq 1$  egész számok.

$m_0, m_1, m_2, m_3, \dots, m_n$  az egyes alappontokhoz tartozó kényszerfeltételek száma. Adottak továbbá az  $f_0^{(0)}, \dots, f_0^{(m_0-1)}, \dots, f_n^{(0)}, \dots, f_n^{(m_n-1)}$  értékek, ahol  $f_i^{(m_i-1)}$  a keresett függvény  $m_i - 1$ -dik deriváltjának az  $x_i$  helyen vett értéke. Legyen  $m = \sum_i m_i$ .

Feladatunk egy olyan legfeljebb  $(m-1)$ -edfokú polinomot keresni, jelölje ezt  $p$ , melyre:

$$p^{(k)}(x_j) = f_j^{(k)}, \quad \text{ahol } k = 0, 1, 2, \dots, m_j - 1 \text{ és } j = 0, 1, 2, \dots, n$$

Vagyis az egyes alappontokhoz nemcsak függvényértékeket, hanem derivált értékeket is rendelünk. Mindezek mellett még azt sem engedjük meg, hogy akár egy derivált is kimaradjon. Tehát nincsen olyan, hogy csak a hatodik derivált adott.

Azt állítjuk, hogy ilyen  $p$  polinom egyértelműen létezik. Jelölje ezt a polinomot a továbbiakban  $H_{m-1}$ . Ezt hívjuk Hermite – polinomnak.

Eddig annyi a különbség az előbbi interpolációs módszerekhez képest, hogy míg azoknál csak az alappontok és a hozzájuk tartozó függvényértékek, addig itt ezen felül még a deriváltak is adottak. Ezzel az új feltétellel a kapott görbénk valóságosabb lesz, új pontok hozzá vételével, illetve az alappontok mozgásával sem romlik el annyira, mint a Lagrange illetve Newton esetén.

A  $p$  polinomot a következő alakban keressük.

$$p(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_{m-1} \cdot x^{m-1}$$

Jelöljük  $\underline{a}$ -val az  $a_0, a_1, a_2, \dots, a_{m-1}$  által meghatározott együtthatóvektort ( $\underline{a} \in R^m$ ),  
 $\underline{a}$  elemeire összesen  $m_0 + m_1 + m_2 + \dots + m_n = m$  darab feltétel van. ( $p^{(k)}(x_j) = f_j^{(k)}$ )

Mátrixokkal és vektorokkal az egyenletrendszer a következőképpen néz ki:

$$A \cdot \underline{a} = \underline{b} \quad , \quad \text{ahol } \underline{b} \in R^m, A \in R^{m \times m}$$

Az, hogy a polinom egyértelműen létezik ekvivalens azzal, hogy a fenti egyenletrendszernek egyértelműen létezik megoldása. Ami pedig ekvivalens azzal, hogy az  $A \cdot \underline{a} = 0$  homogén egyenletnek csak triviális megoldása van.

Alkalmazzuk az ekvivalenciát visszafelé:

Tegyük fel, hogy  $f_j^{(k)} = 0$ . Ekkor olyan polinomot keresünk, melyre teljesül, hogy:

$$p^{(k)}(x_j) = 0 \text{ minden } k \text{-ra és } j \text{-re, ahol a } k = 0, 1, 2, \dots, m_j - 1 \quad j = 0, 1, 2, \dots, n.$$

Meg kell mutatnunk, hogy ebben az esetben  $p \equiv 0$  szükségképpen.

A fenti  $p^{(k)}(x_j) = 0$  feltételből adódik, hogy  $p$ -nek  $x_j$  legalább  $m_j$ -szeres gyöke minden  $j$ -re.  $p$ -nek így multiplicitással együtt legalább  $m_0 + m_1 + m_2 + \dots + m_n = m$  darab gyöke van. Mivel azonban  $p$  legfeljebb  $(m-1)$ -edfokú, ezért  $p \equiv 0$  és így  $a_i = 0 \forall i$ -re, ahol  $i = 0, 1, 2, \dots, n$ .

Megjegyzés:

(i) Ha csak a függvényértékek adottak, az alappontokban, azaz  $m_0 = m_1 = m_2 = \dots = m_n = 1$ , akkor a Lagrange – féle interpolációs polinomot kapjuk vissza, vagyis:

$$H_n(x) = L_n(x)$$

(ii) Ha csak egy alappontunk van, akkor a feladat megoldása a

$$H_n(x) = f_0 + \frac{f_0'}{1!} \cdot (x - x_0) + \dots + \frac{f_0^{(m_0-1)}}{(m_0-1)!} \cdot (x - x_0)^{(m_0-1)}$$

Taylor-polinom, ahol  $n = m_0 - 1$ .

(iii) Hermite – Fejér interpoláció:

Ha  $m_0 = m_1 = m_2 = \dots = m_n = 2$ , azaz az alappontokban a függvényértékek és azok deriváltjai vannak megadva, akkor a  $H_n(x)$  polinom felírható explicit alakban:

$$H_n(x) = \sum_{k=0}^n f_k \cdot \left[ 1 - \frac{\omega''(x_k)}{\omega'(x_k)} \right] \cdot l_k^2(x) + \sum_{k=0}^n f_k' \cdot (x - x_k) \cdot l_k^2(x), \text{ ahol}$$

$$l_k(x) = \frac{\omega(x)}{(x - x_k) \cdot \omega'(x_k)} \quad k = 0, 1, 2, \dots, n \quad \text{és} \quad \omega(x) = (x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_n)$$

Az Hermite interpoláció egyik legalapvetőbb típusa a kétpontos Hermite interpoláció.

### 1.3.1 Kétpontos Hermite – interpoláció

Legyen adott egy  $[a, b]$  intervallum. Az alappontjaink legyenek  $x_0$  és  $x_1$  úgy, hogy  $x_0 = a, x_1 = b$ , továbbá  $n = 1$  és  $m_0 = m_1 = 2$ . Ekkor  $m = 4$ , amiből következik, hogy  $p$  eleme  $P^3$ -nak, vagyis  $p$  egy legfeljebb harmadfokú polinom. A kérdés az, hogy hogyan számoljuk ki  $H_3(x)$ -et.

Az alappontok mellett még adottak a hozzájuk tartozó függvényértékek és azok első deriváltjai. Tehát azt kell belátnunk, illetve olyan polinomot keresünk, hogy a következő teljesüljön:

$$H_3(x_j) = f_j \quad j = 0, 1$$

$$H_3'(x_j) = f_j' \quad j = 0,1$$

A  $H_3(x)$  polinomot általános alakban a következőképpen írhatjuk fel:

$$H_3(x) = A + B \cdot x + C \cdot x^2 + D \cdot x^3$$

Helyette vehetjük a következő, vele ekvivalens alakot is:

$$H_3(x) = A + B \cdot \frac{x-a}{h} + C \cdot \frac{(x-a)^2}{h^2} + D \cdot \frac{(x-a)^3}{h^3}, \text{ ahol } h = b-a.$$

Vegyük ennek az egyenletnek a deriváltját:

$$H_3'(x) = B \cdot \frac{1}{h} + 2 \cdot C \cdot \frac{(x-a)}{h^2} + 3 \cdot D \cdot \frac{(x-a)^2}{h^3}$$

Így az interpolációs feltételek a következőképpen írhatók fel:

$$H_3(x_0) = A = f_0$$

$$H_3(x_1) = A + B + C + D = f_1$$

$$H_3'(x_0) = \frac{B}{h} = f_0' \quad \Rightarrow \quad B = h \cdot f_0'$$

$$H_3'(x_1) = \frac{B + 2 \cdot C + 3 \cdot D}{h} = f_1' \quad \Rightarrow \quad f_1' \cdot h = B + 2 \cdot C + 3 \cdot D$$

Ebből az egyenletrendszerből az  $A, B, C, D$  együtthatókra a következőt kapjuk:

$$A = f_0$$

$$B = h \cdot f_0'$$

$$C = -3 \cdot f_0 + 3 \cdot f_1 - 2 \cdot h \cdot f_0' - h \cdot f_1'$$

$$D = 2 \cdot f_0 - 2 \cdot f_1 + h \cdot f_0' + h \cdot f_1'$$

Ebből pedig már könnyen megkapjuk a  $H_3(x)$ -et, amit kerestünk.

## 2 Spline – interpoláció

Ebben a fejezetben az egyik legelterjedtebb interpolációs eljárással fogunk foglalkozni. Ha az interpolációs alappontok szabadon választhatók, akkor láthattuk, hogy az előzőekben használt módszerek stabil, közelítő polinomokhoz vezetnek. Van azonban olyan, amikor ezeket a pontokat nem lehet szabadon megválasztani. A hétköznapi életben többnyire ilyen esetekkel találkozunk, mivel méréseket nem tudunk akárhol végezni, akár a környezet, az időjárás, vagy egyéb okok miatt.

Az a módszer, hogy egyetlen interpolációs polinommal közelítünk nem feltétlenül jó megoldás, már csak azért sem, mert ebben az esetben nem lehet figyelmen kívül hagyni az alappontok elhelyezkedését a numerikus instabilitás és a rosszul kondicionáltság miatt.

Egy másik módszer a szakaszonkénti polinomiális interpoláció használata. Ebben az esetben viszont a függvény simaságát nem lehet figyelmen kívül hagyni. Ha erről semmit sem tudunk, akkor nem is biztos, hogy érdemes ezt a módszert alkalmazni. Az sem elhanyagolható tényező, hogy akármilyen magas fokú polinomot is alkalmazunk, az egyes szakaszokon, a teljes intervallumon az így kapott görbe első deriváltjainak az alappontokban szakadása lehet, minek következtében a görbénken töréspontok keletkeznek. Ez kifejezetten rossz, főleg ha autót, vagy hajót, vagy akármilyen másik járművet szeretnénk tervezni.

Ekkor jön képbe a spline interpoláció, melynek segítségével olyan interpolációs függvényt lehet előállítani, amely teljesíti a simasági kritériumokat is.

„Spline” az angol neve annak az acélvonalzónak, melyet hajlíthatósága miatt a tervezők, műszaki rajzolóknak nagy előszeretettel használtak adott  $(x_i, y_i)$  pontok összekötéséhez.

Az így kapott görbe simaságát az acélvonalzó törésmentessége biztosítja.

A spline - interpoláció alapja az, hogy az egész intervallumot részintervallumokra osztjuk, a részintervallumokban pedig alacsony fokszámú polinomokat használunk. A megfelelő eredmény érdekében még annak is teljesülnie kell, hogy az intervallumhatároknál a polinomok illesztése folyamatos legyen. Tehát több alacsony fokszámú polinomból összerakott függvényt keresünk úgy, hogy az adott pontokon való áthaladás megkövetelése mellett a polinomok a szomszédos intervallumok csatlakozási pontjaiban előírt differenciálhatósági feltételnek is eleget tegyenek.

A közelítéshez igénybevett polinomok fokszáma alapján lehet szó kvadratikus vagy köbös spline- ról. A korábbi interpolációs eljárásoknál láthattuk, hogy a polinom fokszáma  $n = m - 1$ , ahol  $m$ -mel jelöljük az alappontok számát. Spline- ok alkalmazása esetén a fokszám lényegesen kisebb, mint az alappontok száma.

Spline-on egy szakaszosan parametrikus polinomokkal leírt görbét értünk. Az  $n$ -edfokú spline-ban legfeljebb  $n$ -edfokú polinomszakaszok csatlakoznak egymáshoz, biztosítva a folytonosságot és az  $n-1$ -szeri differenciálhatóságot is. Tehát vegyünk egy  $[a, b]$  intervallumot és definiáljunk ezen egy spline függvényt, legyen ez  $S$ .  $S$ -re teljesül, hogy az  $[a, b]$  intervallum következő felosztását véve :

$$a = x_0 < x_1 < x_2 < x_3 < x_4 < \dots < x_n = b$$

az összes  $[x_i, x_{i+1}]$  részintervallumra megszorítva polinom ezeket jelöljük  $S_i$ -vel és ezen felül a meghatározáshoz a felosztás belső pontjaiban  $(x_1, \dots, x_{n-1})$  csak függvényértékek szükségesek.

Az előbbieken definiált  $S$  spline függvényről a következők mondhatók el:

- (i)  $S(x_i) = y_i$ , vagyis  $S$  áthalad az  $(x_i, y_i)$  koordinátájú pontokon
- (ii)  $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ , ami a folytonosságot jelenti
- (iii)  $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$ , tehát  $S$  - nek nincsenek törései
- (iv)  $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$ , vagyis folytonos a görbülete
- (v) Peremfeltételek: (a) Természetes – peremfeltétel:  $S''(a) = S''(b) = 0$   
 (b) Hermite – peremfeltétel:  $S'(a) = f'(a)$   $S'(b) = f'(b)$   
 (c) Periodikus – peremfeltétel:  $S'(a) = S'(b)$   $S''(a) = S''(b)$   
 ( $f$  a közelítendő függvény,  $a = x_0, b = x_n$ )

(vi)  $O(n)$  műveletigénnyel előállítható

A peremfeltételek alapján különböztetjük meg az interpolációs eljárásokat. A továbbiakban részletesebben foglalkozunk a természetes köbös spline interpolációval.

## 2.1 Természetes köbös spline – interpoláció

Ahogy az a nevében is szerepel itt a természetes – peremfeltételt fogjuk használni. Legyen az  $[x_i, x_{i+1}]$  intervallumon a harmadfokú polinom:

$$S_i(x) = a_{i0} + a_{i1} \cdot x + a_{i2} \cdot x^2 + a_{i3} \cdot x^3$$

Tehát feladatunk  $n$  darab harmadfokú polinomot meghatározni. Ez  $4 \cdot n$  darab ismeretlent jelent (minden polinomban 4 ismeretlen van).

Az interpolációs feltétel,  $f(x_i) = y_i$ , a következőképpen írható fel:

$$S(x_i) = y_i \quad , \text{ ami } n+1 \text{ darab feltétel}$$

További feltétel, hogy  $S, S', S''$  folytonos a belső pontokban, ami összességében  $3 \cdot (n-1)$  darab további feltétel. Összességében tehát eddig  $4 \cdot n - 2$  feltételünk van.

A maradék két feltételt jelen esetben a végpontokra előírtak adják, miszerint:

$$S''(a) = S''(b) = 0$$

A spline- t az  $S''(x_i) = M_i$ ,  $i = 0, 1, 2, \dots, n$  momentumai segítségével állítjuk elő.  
(A momentumok használatának alapja a következő mechanikai megfontolás: egyenes rúd hajlításakor az úgynevezett rugalmas szál alakját meghatározó  $f(x)$  függvény eleget tesz a közelítőleg érvényes  $f'' = \frac{M}{IE}$  differenciálegyenletnek, ahol  $M$  a rúdat terhelő hajlítónyomaték,  $I$  keresztmetszet másodrendű nyomatéka a keresztmetszet súlypontjára vonatkoztatva és  $E$  a rugalmassági modulusz.)

Jelöljük az alappontok közti távolságokat a következő módon:

$$h_{i+1} := x_{i+1} - x_i \quad i = 0, 1, 2, \dots, n-1$$

Az  $S$  függvény második deriváltja szakaszonként lineáris függvény. Ha  $x$  eleme az  $[x_i, x_{i+1}]$  intervallumnak, akkor az  $S''(x)$  a következőképpen írható fel a momentumai segítségével:

$$S''(x) = M_i \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} + M_{i+1} \cdot \frac{x - x_i}{x_{i+1} - x_i}.$$

Ez nem más, mint az elsőfokú Lagrange interpolációs polinom.

$S''(x)$ -ba az előzőekben definiált  $h_i$  hosszakat helyettesítve a következőt kapjuk:

$$S''(x) = M_i \cdot \frac{x_{i+1} - x}{h_{i+1}} + M_{i+1} \cdot \frac{x - x_i}{h_{i+1}}$$

Ezt kétszer  $x$  szerint, ha integráljuk, akkor a következőket kapjuk:

$$S'(x) = -M_i \cdot \frac{(x_{i+1} - x)^2}{2 \cdot h_{i+1}} + M_{i+1} \cdot \frac{(x - x_i)^2}{2 \cdot h_{i+1}} + A_i$$

$$S(x) = M_i \cdot \frac{(x_{i+1} - x)^3}{6 \cdot h_{i+1}} + M_{i+1} \cdot \frac{(x - x_i)^3}{6 \cdot h_{i+1}} + A_i \cdot (x - x_i) + B_i, \text{ ahol } A_i \text{ és } B_i \text{ integrációs}$$

állandók.

Ezen felül tudjuk, hogy  $S(x_i) = y_i$ , amiből  $M_i \cdot \frac{(h_{i+1})^2}{6} + B_i = y_i$ . Átrendezéssel  $B_i$ -re azt kapjuk, hogy:

$$B_i = y_i - M_i \cdot \frac{h_{i+1}^2}{6}$$

Az interpolációs feltételt  $x_{i+1}$  pontban is felírjuk, vagyis:

$$S(x_{i+1}) = y_{i+1}$$



Ebből  $y_{i+1}$ -re kapjuk, hogy  $M_{i+1} \cdot \frac{h_{i+1}^2}{6} - A_i \cdot h_{i+1} + y_i - M_i \cdot \frac{h_{i+1}^2}{6}$ , amiből pedig  $A_i$  értéke is meghatározható, tehát:

$$A_i = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{6} \cdot (M_{i+1} - M_i)$$

Az egyenletekben már csak az  $M_i$  momentumok az ismeretlenek. Ezeket az  $S$  deriváltjának a belső csomópontokbeli folytonosságából határozhatjuk meg:

$$S'(x_i^+) = S'(x_i^-), \quad \text{ahol } i = 1, 2, \dots, n-1$$

Két egymáshoz kapcsolódó intervallumra felírjuk  $S_i$  deriváltját. Vegyük tehát a következő két egyenletet:

$$x \in [x_{i-1}, x_i]: \quad S_i'(x) = -M_{i-1} \cdot \frac{(x_i - x)^2}{2 \cdot h_i} + M_i \cdot \frac{(x - x_{i-1})^2}{2 \cdot h_i} + \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{6} \cdot (M_i - M_{i-1})$$

$$x \in [x_i, x_{i+1}]: \quad S_i'(x) = -M_i \cdot \frac{(x_{i+1} - x)^2}{2 \cdot h_{i+1}} + M_{i+1} \cdot \frac{(x - x_i)^2}{2 \cdot h_{i+1}} + \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{6} \cdot (M_{i+1} - M_i)$$

Ebből:

$$S'(x_i^-) = \frac{y_i - y_{i-1}}{h_i} + \frac{h_i}{3} \cdot M_i + \frac{h_i}{6} \cdot M_{i-1} \quad i = 1, 2, 3, \dots, n-1$$

$$S'(x_i^+) = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{3} \cdot M_i - \frac{h_{i+1}}{6} \cdot M_{i+1} \quad i = 1, 2, 3, \dots, n-1$$

Ha felírjuk az ezek egyenlőségére vonatkozó egyenletet, akkor  $M_i$ -re a következő  $(n-1)$  darab egyenletből álló rendszer jön ki:

$$\frac{h_i}{6} \cdot M_{i-1} + \frac{h_i + h_{i+1}}{3} \cdot M_i + \frac{h_{i+1}}{6} \cdot M_{i+1} = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \quad i = 1, 2, \dots, n-1$$

Szorozzuk meg a fenti egyenletet  $\frac{6}{h_{i+1} + h_i}$  - vel. Vezessük be a következő jelöléseket:

$$\mu_i = \frac{h_i}{h_{i+1} + h_i}, \quad \lambda_i = \frac{h_{i+1}}{h_{i+1} + h_i}, \quad d_i = \frac{6}{h_{i+1} + h_i} \cdot \left( \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right)$$

A fentiek behelyettesítésével a következőket kapjuk:

$$\mu_i \cdot M_{i-1} + 2 \cdot M_i + \lambda_i \cdot M_{i+1} = d_i \quad i = 1, 2, 3, \dots, n-1$$

0- ra és  $n$ - re legyen  $\lambda_0 = d_0 = \mu_n = d_n = 0$ . Így  $M_i$ - re a következő lineáris egyenletrendszert kapjuk :

$$\begin{aligned} 2 \cdot M_0 + \lambda_0 \cdot M_1 &= d_0 \\ \mu_1 \cdot M_0 + 2 \cdot M_1 + \lambda_1 \cdot M_2 &= d_1 \\ &\vdots \\ \mu_{n-1} \cdot M_{n-2} + 2 \cdot M_{n-1} + \lambda_{n-1} \cdot M_n &= d_{n-1} \\ \mu_n \cdot M_{n-1} + 2 \cdot M_n &= d_n \end{aligned}$$

Ez mátrixalakban egy tridiagonális, vagyis háromátlós egyenletrendszer:

$$\begin{bmatrix} 2 & \lambda_0 & 0 & \dots & 0 \\ \mu_1 & 2 & \lambda_1 & \dots & 0 \\ 0 & \mu_2 & 2 & \dots & 0 \\ \vdots & \dots & \dots & \dots & \vdots \\ 0 & \dots & \mu_{n-1} & 2 & \lambda_{n-1} \\ 0 & \dots & 0 & \mu_n & 2 \end{bmatrix} \cdot \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

Ezt az egyenletrendszert rövidített Gauss algoritmussal hatékonyan megoldhatjuk. Ezek után már nincs ismeretlen, vagyis egyszerű behelyettesítéssel már meg is kaptuk a keresett harmadfokú spline- t.

Harmadfokú spline- ok mellett beszélhetünk még másodfokúakról is. Ezeknek azonban nincsen olyan fizikai háttere, mint a köbösnek az acélhúr.

## 2.2 Másodfokú - spline

Adott  $[a, b]$  intervallum és rajta az alappontok, melyek nem feltétlenül ekvidisztánsak és a következőképpen néznek ki:

$$a = x_0 < x_1 < x_2 < \dots < x_n = b$$

Adottak még ezen kívül az egyes pontokhoz tartozó  $f_i$  értékek és az  $f_0'$  értéke is, ahol természetesen  $i = 0, 1, 2, \dots, n$ . Az intervallumok hosszát jelöljük  $h_k$  - val:

$$h_k = x_{k+1} - x_k$$

Ekkor a másodfokú spline – interpoláció nem más, mint egy másodfokú Hermite interpoláció a részintervallumokra alkalmazva.



2.ábra

Ahogy a fenti ábra is mutatja csak a baloldali végpontban ismert a derivált értéke. Így összesen három kényszerfeltételünk van (a két végpontban a függvényértékek és a baloldaliban a derivált értéke).

Ugyanakkor a másodfokú Hermite – interpolációs polinomot a következő egyenletből kaphatjuk meg:

$$H_k^{(2)}(x) = a + b \cdot x + c \cdot x^2$$

Az  $f_k, f_k'$  és  $f_{k+1}$  alapján a fenti egyenlet három ismeretlenje,  $a, b$  és  $c$  nagyon egyszerűen számítható, amiből pedig  $H_k^{(2)}(x)$  előállítható.

Ezek után  $f_{k+1}'$  – at a következőképpen kaphatjuk meg:

$$f_{k+1}' = (H_k^{(2)})'(x_{k+1})$$

Ezzel a módszerrel tehát minden további derivált meghatározható, az  $f_0'$  értékére az iterációs eljárás elindításához volt szükségünk.

Ez az interpolációs eljárás majdnem olyan jó, mint a köbös spline, azzal a különbséggel, hogy, míg ott  $C^2$  – folytonos, itt csak  $C^1$  – folytonos a csatlakozás az alappontokban.

A másod és harmadfokú spline interpoláció mellett természetesen lehetne beszélni magasabb fokú interpolációkról is, de nem érdemes, mivel jelentős javulás nem tapasztalható és több számolást is igényelnek. A legelterjedtebb és legjobban használható a fentiekben bemutatott természetes köbös spline interpoláció.

A következő fejezetben az általam készített program segítségével szemléltetem a természetes köbös spline interpoláció és a Lagrange interpolációs eljárás által adott interpolációs görbe közti különbséget.

## 3 A program

### 3.1 Augmented Reality ( kiterjesztett valóság )

A kiterjesztett valóság (AR) egy viszonylag új, egyre jobban elterjedő technológia, melynek segítségével a virtuális elemek keveredni tudnak a valós fizikai élettel. Ilyen elem lehet például egy három dimenziós modell, vagy akár egy animáció. Az AR definiálására több különböző definíció is létezik, azonban a legszélesebb körben elfogadott Ronald T. Azuma nevéhez fűződik, mely szerint a kiterjesztett valóság valós időben, akár interaktívan ötvözi a valós és virtuális világot. Az így kapott rendszer a valós és virtuális világ között helyezkedik el. Ennek megjelenítése számos eszköz segítségével történhet, a legáltalánosabb, hogy egy monitoron vagy egy telefon kijelzője.

A kiterjesztett valóság kétféle típusát különböztethetjük meg a megjelenítés és a virtuális elem helyzete függvényében.

Az első ilyen típus a pozíció és irány alapú, melyet elsősorban mobiltelefonokon alkalmaznak. Célja, hogy a kijelzőn megjelenő valós képet új információkkal egészítse ki. A valós képet kiterjesztő információ helyének meghatározásához a készülék iránytűjét, a GPS pozícióját, valamint gyorsulásérzékelőjét használja a program. A GPS pozíció meghatározza a mobil eszköz pontos helyzetét, ami alapján kiszámolható, hogy az illető milyen messze is van az “új információtól”.

Jelenleg több ilyen alkalmazás is létezik, ilyen a Wikitude (<http://www.wikitude.org>), a Layar (<http://www.layar.com/>), valamint a Junaio (<http://www.junaio.com/>).



3. ábra: A pozíció és irány alapú AR okostelefonon való megjelenése [Wikitude](#) segítségével

A kiterjesztett valóság másik típusa a marker alapú. Egy speciális képrészletet keresünk, amely kitűnik a környezetéből, így könnyen kereshető és megtalálható. Ajánlott hibatűrő kódolást alkalmazni, csökkentve a valószínűségét a hibás marker detektálásnak. Az interneten számos leírás van arról, hogy milyen további kritériumoknak kell megfelelnie a markereknek. Azonban ha nem akarunk ilyen mélyen belemenni a részletekbe, akkor elég csak letölteni egyet, majd kinyomtatni. A későbbiekben bemutatásra kerülő programban a következő markert használtam.



4.ábra: A program által használt marker

A marker két fő részből áll: egy négyzetből, tetszőleges vastagságú szegéllyel, amelyen belül egy egyedi, fekete-fehér képrészlet található. A mai technológia lehetővé teszi hogy a fenténél látványosabb markereket használjunk, például a következőt a NINTENDO játékszoftver és konzol gép gyártó cég készítette egyik játékához.



5.ábra: NINTENDO által használt AR marker

Sikeres detektálást követően a marker helyére illeszthető a virtuális elem, mely a marker mozgásából kiszámítja az elvégzendő transzformációkat, így ötvözve a képernyőn a virtuális valóságot az igazi valósággal.

A kiterjesztett valóság egyre inkább megjelenik hétköznapjainkban, néhány gyakorlati alkalmazás:

- oktatás ([http://www.learnar.org/bio\\_organs\\_demo.html](http://www.learnar.org/bio_organs_demo.html))
- szerelés (<http://www.youtube.com/watch?v=P9KPJIA5yds>)
- reklám  
([http://www.youtube.com/watch?v=RnN6s0xfMvs&feature=player\\_embedded#at=40](http://www.youtube.com/watch?v=RnN6s0xfMvs&feature=player_embedded#at=40))
- sport (<http://www.youtube.com/watch?v=DnmXT6x85p8&feature=fvst>)
- játék (<http://www.youtube.com/watch?v=Lfp8id6bpDU>)
- marketing (<http://www.youtube.com/watch?v=NxQZuo6pFUw>)

### 3.2 A program részletes ismertetése

Dolgozatomhoz egy marker alapú AR alkalmazást készítettem, az FLARToolKit segítségével. A programozási nyelv actionscript 3, a fejlesztői környezet pedig Flash CS4. Az actionscript 3 egy objektumorientált programozási nyelv, mely a hasonlóaktól, mint például a C#-tól csak szintaktikailag tér el. A program célja részben az AR kódok “világának” megismertetése, illetve a technológia segítségével a Lagrange és a Természetes köbös spline interpoláció szemléltetése, ez utóbbit 3D-ban is. A 3D-s tartalmak legyártásához a papervision3d nyílt forráskódú könyvtárat használtam, melyhez az ábrán lévő marker csatolva volt. A markerhez pattern fájlra van szükség, mely vagy adott, vagy magunknak kell elkészíteni. Mivel én nem magam készítettem, így ebbe nem is mennék bele, minden fejlesztői eszköznél megtalálható a leírás, hogy hogyan kell. Az alkalmazás a pattern fájlra köszönhetően fogja majd tudni transzformálni a markerhez tartozó grafikus elemeket.

A programot a könnyebb futtathatóság kedvéért html-be exportáltam, így az indításhoz csak meg kell nyitni egy böngészőben, amely engedélyezi a flash player futását. (Természetesen a html kód mellett még elkészül a flash .fla fájljából fordított .swf is, melynek megnyitása azonban egyes gépeken tapasztalataim alapján viszonylag bonyolultnak bizonyult és további programok letöltését igényelte, melyekre így nincsen szükség.)

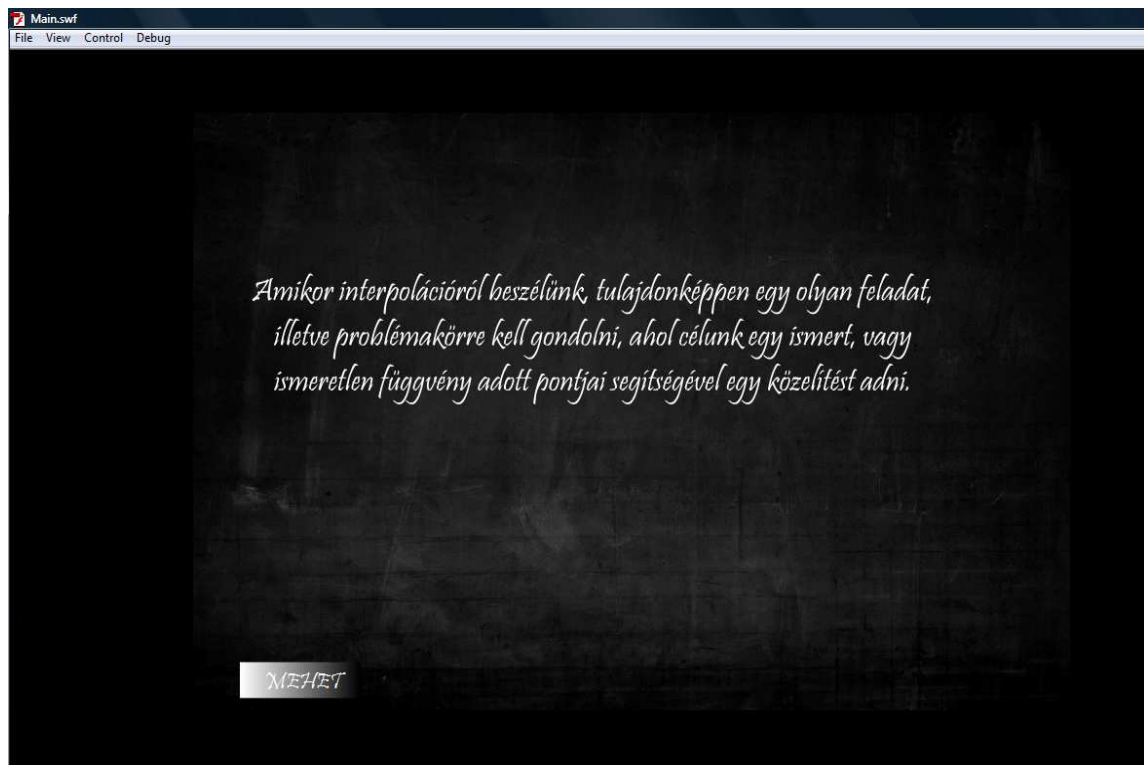
Indításkor feljön egy köszöntő ablak és leírja a program rövid feladatát, majd választhatunk, hogy két vagy három dimenziós interpolációt szeretnénk látni.





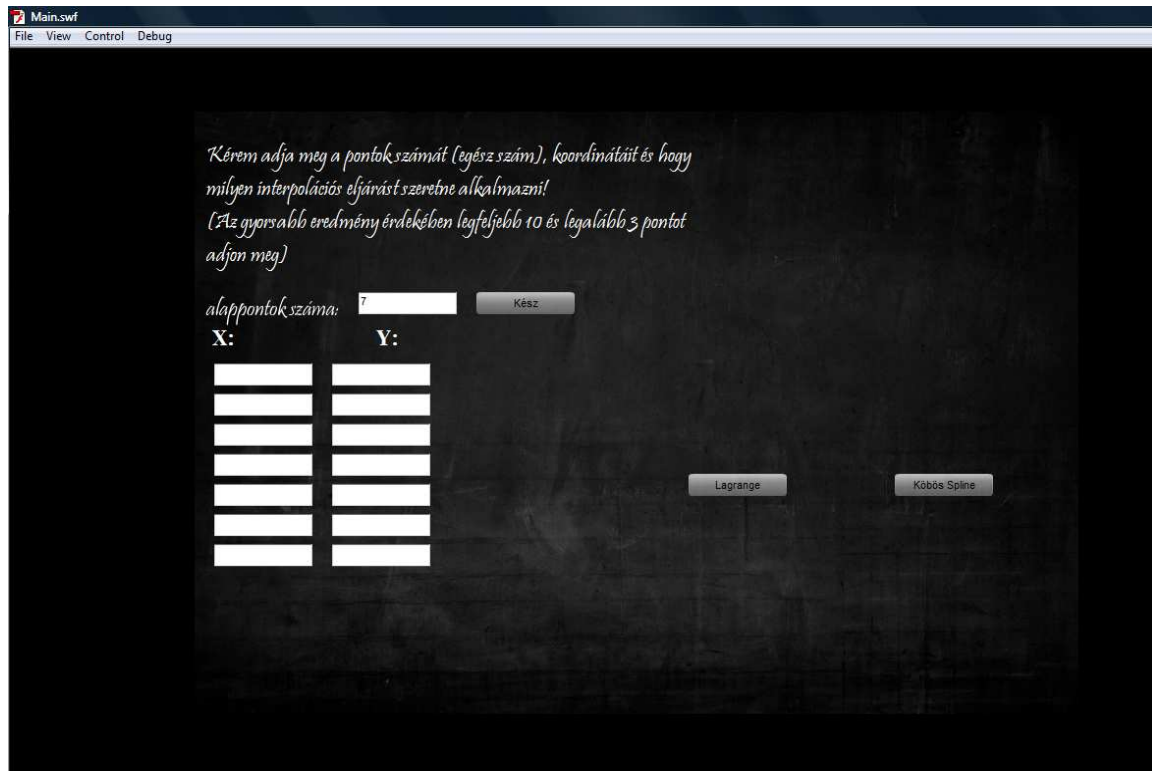
6. ábra: Köszöntő ablak

A megfelelő gombra kattintva a következő oldalon a választásunktól függetlenül egy rövid leírást találunk az interpolációs eljárásokról, illetve a „Mehet” gombra kattintva továbbléphetünk.



7. ábra: Az interpolációs eljárás meghatározása

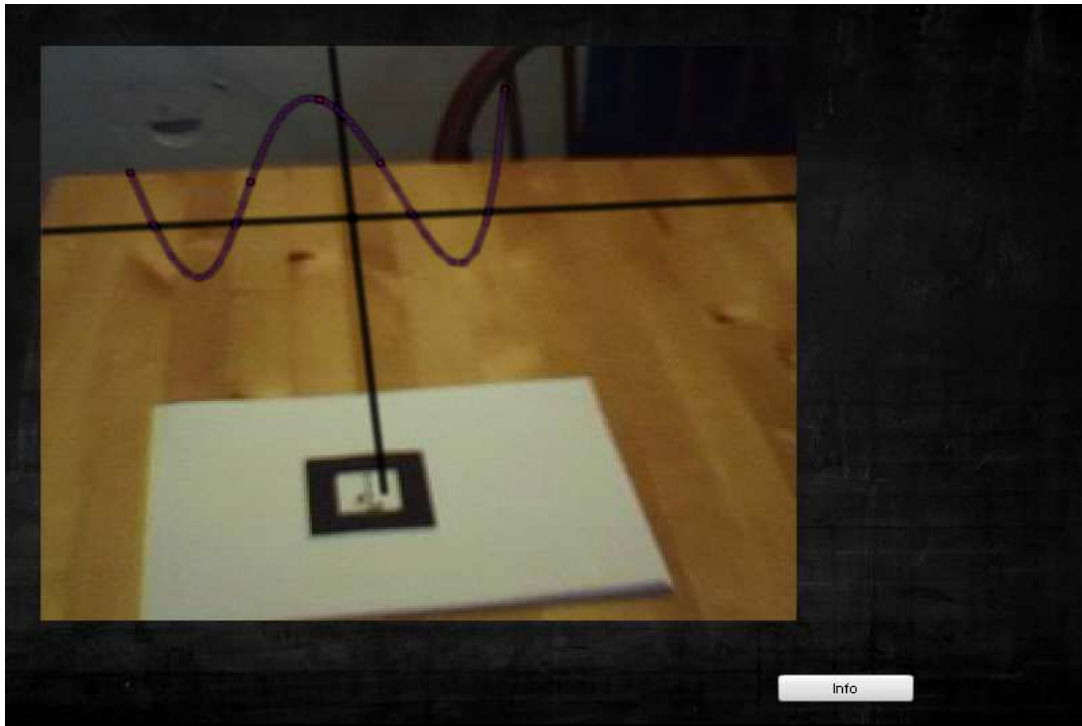
A következőekben megadhatjuk, hogy hány alappontot szeretnénk és hogy mik legyenek ezek koordinátái. A gyorsabb eredmény végett az alappontok száma korlátozva van, legalább három és legfeljebb tíz lehet. Az inputmezőktől jobbra látható a két, vagy egy gomb, attól függően, hogy hány dimenziós interpolációt nézünk.



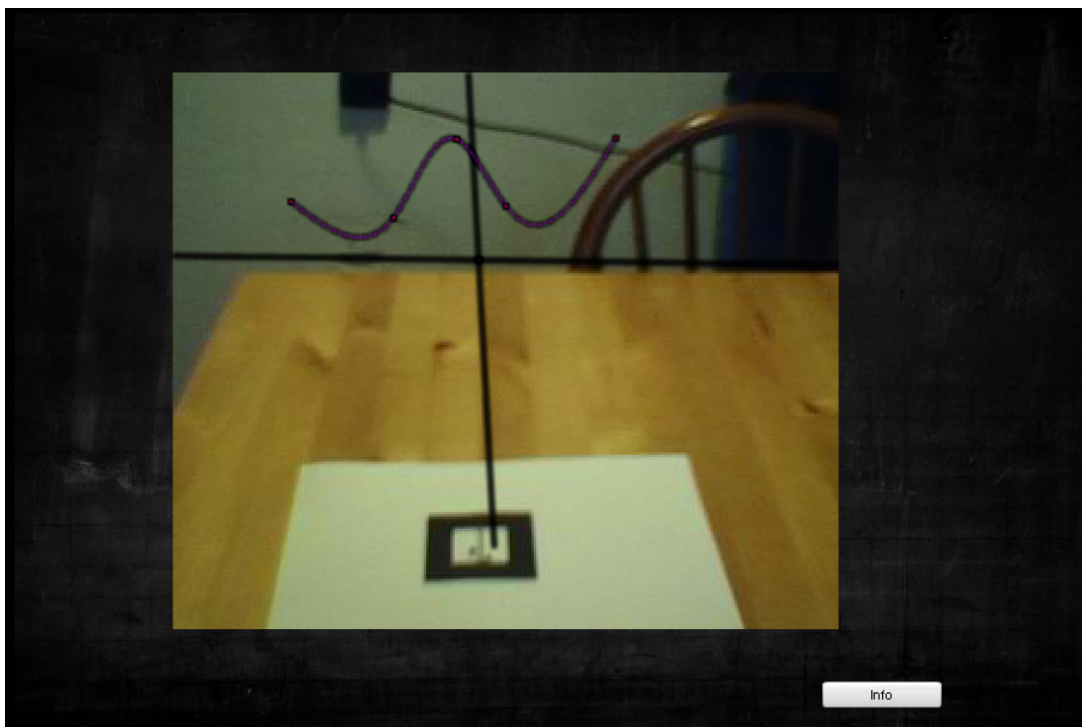
8. ábra: Bemenő paraméterek megadása

Ezek valamelyikére kattintva, majd a webkamera program általi használatának engedélyezése után a markert a kamera elé tartva láthatjuk is a kirajzolt képet.

Az alábbi két ábra az ugyanazon pontokra lefuttatott két dimenziós Lagrange és természetes köbös spline interpolációs eljárás eredményét mutatja.

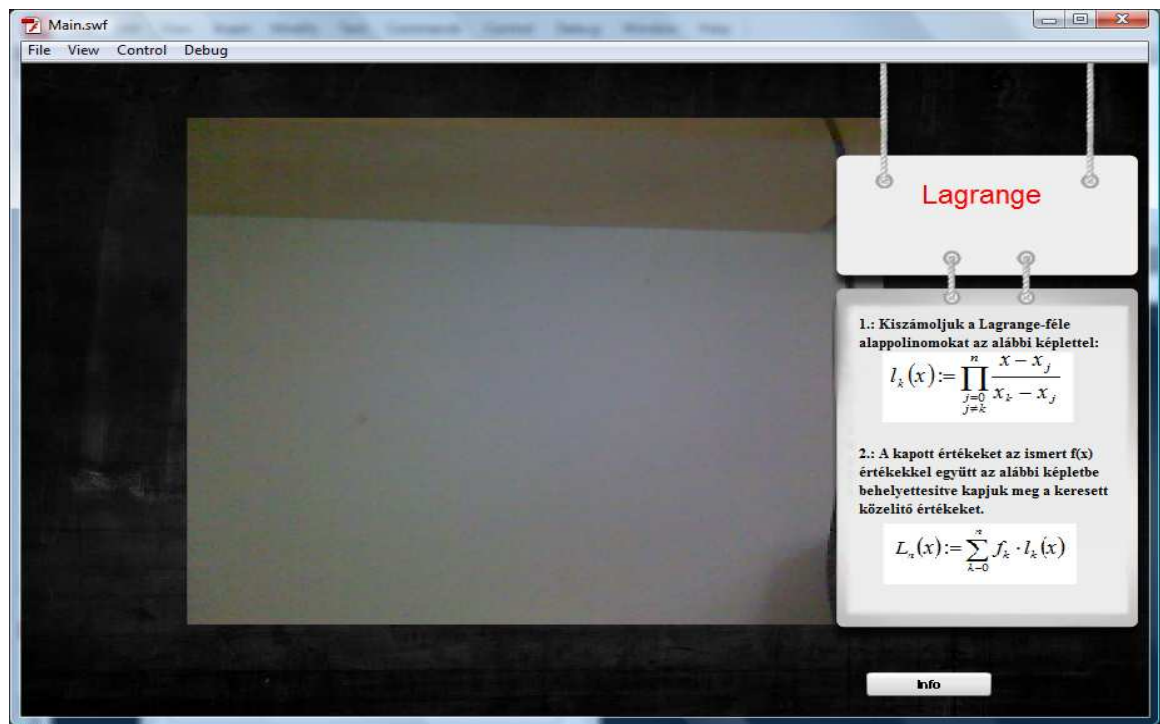


9. ábra: Lagrange interpoláció eredménye



10. ábra: Természetes köbös spline interpoláció eredménye

Az „infó” gombra kattintva ismerkedhetünk meg, a monitorra kirajzolt függvénygörbe matematikai, numerikus analízisbeli hátterével.



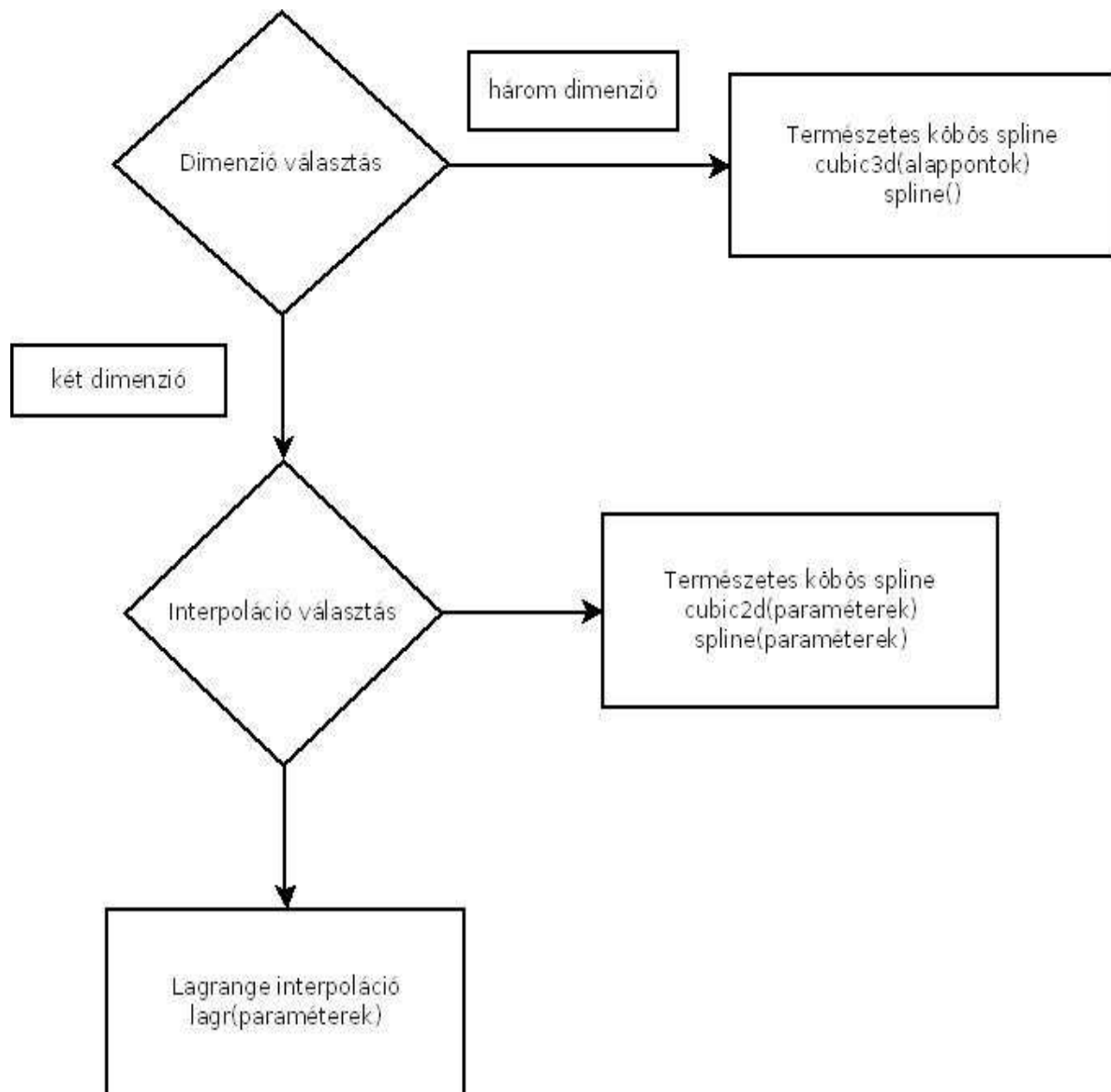
11. ábra: A közelítő függvények megadása

Ezt bezárva választhatunk, hogy újraindítjuk-e a programot esetleg egy másik interpolációs eljárás megismerése végett.

## Összefoglalás

Dolgozatom célja a függvényközelítések, azon belül is az interpolációs eljárások ismertetése. Ez úton is szeretném megköszönni témavezetőmnek, Dr. László Lajosnak, a dolgozatommal kapcsolatos észrevételeit és megjegyzéseit. Az eljárások gyakorlati alkalmazásának bemutatásához az általam készített programban igyekeztem a legújabb és leglátványosabb technológiát használni. Ezen technológia iránti érdeklődésemet Turcsányi-Sz. Márta egyetemi docens egyik prezentációja keltette fel. A program elkészítésekor az internetes dokumentációkra hagytam. Az olvasó a dolgozat végén feltüntetett irodalomjegyzék segítségével jobban beleáshatja magát a közelítő eljárásokba, az approximációs eljárásokba, melyekről itt nem esett szó, és az AR kódok világába. A mellékletben feltüntetett kódrészletek és a folyamatábra a program működésének megértéséhez segítség.

## Melléklet



12.ábra: Folyamatábra

### A használt függvények leírása

lagr(paraméterek):

paraméter: az alappontok x koordinátái

feladata: kiszámítani a lagrange alappolinomokat

cubic3d/2d(alappontok/paraméter):

paraméter: az alappontok koordinátái

feladata: kiszámítani a spline-hoz szükséges paramétereket (a 3-adjokú polinom együtthatóit) és ezekkel meghívni a spline(paraméterek) függvényt, ami

kiszámítja az ábrázoláshoz szükséges további belső pontokban a közelítő függvényértéket

## Kódrészletek

Lagrange alappolinomokat meghatározó függvény:

```
public function lagr(parametr,pol)
{
    var fx = 0;
    for(var i=0; i<egyenes.length; i++)
    {
        var Lg = 1;
        for(var j=0; j<egyenes.length; j++)
        {
            if(i != j)
            {
                Lg *= (parametr-egyenes[j])/(egyenes[i]-egyenes[j]);
            }
        }
        fx += Lg*kettes[i];
    }
    pol.push(fx);
}
```

A spline együtthatóinak kiszámítása:

```
for(var j=0;j<=N-1;j++)
{
    h[j+1]=elso[j+1]-elso[j];
}
for(var l=1;l<=N-1;l++)
{
```

```

sub[l]=h[l]/(h[l+1]+h[l]);
sup[l]=h[l+1]/(h[l+1]+h[l]);
d[l]=6/(h[l+1]+h[l])*((masodik[l+1]-masodik[l])/h[l+1]-(masodik[l]-masodik[l-
1])/h[l]);
}
for(var f=0;f<np;f++)
{
    diag[f]=2;
}
sup[0]=sub[N]=d[N]=d[0]=0;
solveTridiag(sub,diag,sup,d,N,M);
for(var o=0;o<N;o++)
{
    A[o]=(masodik[o+1]-masodik[o])/h[o+1]-h[o+1]/6*(M[o+1]-M[o]);
    B[o]=masodik[o]-M[o]*h[o+1]*h[o+1]/6;
}
for(var q=0;q<egyenes.length-1;q++)
{
    var lepeskoz=h[q+1]/10;
    for(var p=0;p<10;p++)
    {
        var ujk=Number(egyenes[q])+p*Number(lepeskoz);
        ujk.push(ujk);
        var x0=egyenes[q];
        spline(ujk,egyenes,x0,M,A,B,h,ujyok,kettes)
    }
}

```

A spline függvény :

```

public function spline(elem,tomb,x0,momentum,intA,intB,dist,ujy,yostomb)
{
    var also;
    var felso;

```



```

for(var i=0;i<tomb.length;i++)
{
    if(tomb[i]==x0)
    {
        also=i;
        felso=i+1;
    }
}
var Y=momentum[also]*(((tomb[felso]-elem)*(tomb[felso]-elem)*(tomb[felso]-elem))/(6*dist[felso]))+(momentum[felso]*(((elem-tomb[also])*(elem-tomb[also])*(elem-tomb[also]))/(6*dist[felso])))+(intA[also]*(elem-tomb[also]))+intB[also];
ujy.push(Y)
}

```

## Irodalomjegyzék

Dr. Gáspár Csaba: Numerikus Analízis II. előadás

Peter Henrici: Numerikus Analízis, Műszaki Könyvkiadó, Budapest, 1985

Dr. Bajcsay Pál: Numerikus Analízis, Tankönyvkiadó, Budapest, 1978

Sövegjártó András: Numerikus Analízis II., jegyzet programozó és programtervező matematikus szakos hallgatóknak, 2004

Stoyan Gisbert, Takó Galina: Numerikus módszerek I., ELTE TYPOTEX, 1995

<http://hu.wikipedia.org>

<http://www.libspark.org/wiki/saqoosha/FLARToolKit/en>

<http://www.words.transmote.com/wp/flarmanager/>