

**DEBRECENI EGYETEM  
TERMÉSZETTUDOMÁNYI KAR  
MATEMATIKAI ÉS INFORMATIKAI INTÉZET**

HARANGI GABRIELLA  
*programtervező matematikus hallgató*

**GÖRBEAPPROXIMÁCIÓ ÉS MANIPULÁCIÓK AZ APPROXIMÁLT  
GÖRBÉVEL**

APPROXIMÁCIÓ ÉS INTERPOLÁCIÓ  
KOMBINÁCIÓJA B-SPLINE GÖRBÉK ESETÉN

**DIPLOMAMUNKA**

TÉMAVEZETŐ:  
DR. HABIL SZABÓ JÓZSEF  
*egyetemi docens*



**DEBRECEN  
2003**



Köszönetet mondok témavezető tanáromnak, Dr. Szabó Józsefnek, továbbá Kovács Emődnek, a diplomamunkám elkészítéséhez nyújtott segítségükért.

## TARTALOM:

|   |           |
|---|-----------|
| <b>1. BEVEZETÉS.....</b>                    | <b>6</b>  |
| <b>2. MATEMATIKAI, GRAFIKAI ALAPOK.....</b> | <b>8</b>  |
| 2.1. MÁTRIXOK.....                          | 8         |
| 2.2. PONTTRANSZFORMÁCIÓK .....              | 13        |
| 2.3. GÖRBÉK.....                            | 15        |
| 2.3.1. <i>Hermite-ív:</i> .....             | 16        |
| 2.3.2. <i>Bézier-görbe:</i> .....           | 16        |
| 2.3.3. <i>B-spline görbék:</i> .....        | 19        |
| <b>3. B-SPLINE GÖRBE .....</b>              | <b>22</b> |
| 3.1. A B-SPLINE GÖRBE DEFINÍCIÓJA .....     | 22        |
| 3.2. AZ ALAPFÜGGVÉNY .....                  | 23        |
| 3.3. A GÖRBE TULAJDONSÁGAI.....             | 25        |
| <b>4. APPROXIMÁCIÓ.....</b>                 | <b>28</b> |
| 4.1. ELSŐFOKÚ GÖRBE SZERKESZTÉSE.....       | 28        |
| 4.2. ZÁRT GÖRBE SZERKESZTÉSE .....          | 30        |
| 4.3. CSOMÓÉRTÉK MULTIPLICITÁSA .....        | 30        |
| 4.4. KONTROLLPONTOK MULTIPLICITÁSA.....     | 31        |
| <b>5. INTERPOLÁCIÓ.....</b>                 | <b>33</b> |
| 5.1. INTERPOLÁCIÓS ELJÁRÁSOK .....          | 33        |
| 5.1.1. <i>Lagrange interpoláció</i> .....   | 34        |

|           |   |           |
|-----------|---|-----------|
| 5.1.2.    | <i>Hermite-ív</i> .....                 | 35        |
| 5.1.3.    | <i>Interpolációs spline-ok</i> .....    | 37        |
| 5.2.      | ALKALMAZOTT INTERPOLÁCIÓS MÓDSZER ..... | 40        |
| <b>6.</b> | <b>A KÉT TECHNIKA ÖTVÖZÉSE</b> .....    | <b>47</b> |

## 1. Bevezetés

A számítógépes grafika egyik igen érdekes területe a görbemodellezés. Léteznek approximációs és interpolációs görberajzoló eljárások, de mindkét eljárástípus feladata, hogy a koordinátarendszerben megadott pontok sorozatához illesszen egy olyan görbét, mely adott feltételeknek eleget tesz. Ezen terület fontosságát mi sem mutatja jobban, minthogy milyen széles körben használatosak a görbemodellezés különböző megvalósításai. Példaként említhető például a Microsoft Word, ahol a true type betűtípusok B-Spline segítségével vannak megjelenítve, de az Autocad rendszerekben is fontos szerepet játszanak. Valamint igen jelentős és érdekes szerepe van a görbeinterpolációnak a képfeldolgozás területén is.

Ezen diplomamunka célja egy számítógépes program készítése volt, mely a felhasználó által megadott pontokhoz a kívánságnak megfelelően hozzáilleszt egy görbét. Az illesztés módja lehet approximáció, interpoláció illetve az említett két technika ötvözése. A legtöbb görbetípus esetén az approximációt tekintjük alapértelmezettnek. A B-Spline görbék is approximációs jellegű görbék. Ugyanakkor a gyakorlatban sokszor szükség van az adott pontok interpolálására, így a legtöbb görberajzoló eljáráshoz kifejlesztették az interpolációs változatát is. De miért is van szükség az interpolációs eljárásokra?

Egy általános B-Spline görbe definiálása a kontrollpontok, a csomóértékek és a fokszám megadásával történik. Tehát a hagyományos approximációs eljárás keretében kapunk egy olyan görbét, mely megközelíti az adott pontokat, de csak speciális esetekben fog egy-egy pontot érinteni. A felhasználó szemszögéből nézve a dolgot, sokkal inkább hasznos lenne, ha az adott pontokon átmenne a görbe, hiszen legtöbb esetben az az információ áll a rendelkezésünkre, hogy az adott pont a majdani görbe része. Ez a helyzet az interaktív tervezőrendszerekkel is. Ott is interpolációt alkalmaznak, hiszen egy kívánt

alak előállítására B-Spline görbével úgy történik, hogy megadják azokat a pontokat, melyek megközelítőleg a görbén helyezkednek el, majd a rendszer meghatározza az adott pontokon áthaladó görbét. A későbbiekben a kiszámított kontrollpontok helyzetének módosításával finomítható a kapott görbe. A görbe bármely paraméterének megváltoztatásával szintén módosítható a görbe. A kontrollpontok változtatása talán a legvizuálisabb módszer.

Bizonyos helyzetekben szükség lehet az interpoláció és az approximáció összekapcsolására. Ezen feladat egyik legjellemzőbb területe az autógyártás. Az autók karosszériájának tervezése során szükség van a B-Spline görbe által nyújtott nagyfokú flexibilitásra, hogy minél szebb formájú autókat tervezhessenek. Ugyanakkor néhány pont helyzetének viszonylag rögzítettnek kell lennie. Ilyenek például a hegesztési pontok. Így hát ezen pontokat érdemes interpolálni, míg a többit approximálhatjuk. Itt meg kell említeni, hogy az autógyártás esetében nem igazán görbemodellezésről, hanem felületmodellezésről kell beszélnünk, hiszen egy autó dimenziójából egyértelműen következik, hogy itt nem adott pontsorozatra kell görbét illeszteni, hanem a térben megadott pontrácshoz kellene illesztést végeznünk. Tudjuk viszont, hogy például egy B-Spline felület B-Spline görbék tenzori szorzataként jön létre, tehát az approximáció és interpoláció felületek esetében hasonlóan értelmezhető mint görbék esetén.

A program, mely Delphi 5-ben íródott, lehetőséget biztosít kontrollpontok megadására, azok helyzetének megváltoztatására; választási lehetőséget biztosít approximáció, interpoláció illetve ezek kombinációja között, valamint megtekinthetők illetve megadhatók a görbe egyéb paraméterei is, mint például a fokszám. Ezek megvalósításához természetesen szükségem volt különböző matematikai és grafikai ismeretekre, amelyeket a következő fejezetekben részletesebben kifejtek

## 2. Matematikai, grafikai alapok

Ebben a fejezetben azokat a matematikai és grafikai definíciókat és képleteket szeretném bemutatni, melyekre a programomban szükségem volt. Grafikai háttérként be szeretnék mutatni pár görbemodellezési módszert, azt, hogy időben hogyan alakultak ki ezek, valamint fejlődési szintjeiket. Szó lesz a centrális projekcióról, mely akkor használható, ha a görbepontok 3 dimenziós koordináta-rendszerben adottak, valamint a mátrixműveletekről, melyek nélkülözhetetlenek voltak az interpoláció megvalósításánál. Részletesebb kifejtésükre szolgálnak a következő alfejezetek.

### 2.1. Mátrixok

Lineáris egyenletrendszerek számítógépes megoldása során az egyenletrendszer együtthatóit mátrix alakba rendezzük, s ezután az egyenletrendszer megoldását mátrixműveletekkel végezzük el. Mi is a mátrix?

*Definíció:* Általánosan mátrixnak nevezünk minden olyan  $n*m$  számú  $a_{ik}$  mennyiség alábbiak szerinti téglalap alakú elrendezését, amelyet szögletes vagy kerek zárójelbe tesznek és ahol  $i=1,2, \dots,n$ ;  $k=1,2, \dots,m$ .

$$\begin{bmatrix} a_{11} & a_{12} & \Lambda & a_{1m} \\ a_{21} & a_{22} & \Lambda & a_{2m} \\ & & \mathbf{M} & \\ a_{n1} & a_{n2} & \Lambda & a_{nm} \end{bmatrix}$$

Az előbbi mátrixra azt mondjuk, hogy  $n*m$  típusú, mert  $n$  sorból és  $m$  oszlopból áll, az  $a_{ik}$  mennyiségeket pedig a mátrix elemeinek nevezzük. Kvadrátikus (négyzetes) mátrixnak nevezzük az olyan mátrixot, amely sorainak száma megegyezik az oszlopainak számával. Egy többismeretlenes lineáris egyenletrendszer a következőképpen írhatunk fel:

$$\begin{aligned} y_1 &= a_{11} * x_1 + a_{12} * x_2 + \dots + a_{1m} * x_m \\ y_2 &= a_{21} * x_1 + a_{22} * x_2 + \dots + a_{2m} * x_m \\ &\quad \vdots \\ y_n &= a_{n1} * x_1 + a_{n2} * x_2 + \dots + a_{nm} * x_m \end{aligned}$$

Mátrixos alakban ez így írható fel:

$$[y] = [A] * [x]$$

Ahhoz, hogy egy approximációs jellegű görbét interpolációs görbeként használjunk, még szükségünk van néhány mátrixfogalomra.

**DETERMINÁNS FOGALMA:**

A determinánsok segítségével bonyolult és nehezen áttekinthető kifejezéseket tehetünk könnyen kezelhetővé. A determinánsok értéke számítógépen rendkívül gyorsan meghatározható, így nemcsak áttekinthető, hanem gyorsabb számításokat is eredményez. A determináns jelölésére függőleges vonalakat használunk:

$$|A| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

A determinánsok különböző rendűek lehetnek. A fenti példa egy  $n$ -ed rendű determinánst jelöl. Ennek kiszámítása alacsonyabb rendű determinánsszámoló módszerre vezethető vissza, így hát közelítsük most meg a dolgot a másik végéről, s nézzük meg, hogyan lehet kiszámítani a *másodrendű determinánst*.

A másodrendű determináns esetén a mátrixunk 4 elemből áll, melyek  $a_{11}$ ,  $a_{12}$ ,  $a_{21}$ ,  $a_{22}$ ., jelölése:

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix},$$

ahol  $D = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$

A harmadrendű determináns egy olyan 9 elemből álló mátrix, melynek a következő értékeket tulajdonítjuk:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} = \\ = a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}$$

Látható itt is, hogy ezt a harmadrendű determinánst másodrendű aldeterminánsok segítségével számoltuk ki. A fenti példa a determináns kifejtése a mátrix első sora szerint. Megjegyzendő, hogy 3-ad rendű determináns esetén alkalmazható a Sarrus szabály is, melyet úgy kell elképzelni, hogy a mátrix mellé leírjuk még egyszer az első és a második sort, és így a főátló irányában összeköthető elemhármások szorzatösszegéből kivonjuk a mellékátló irányában összeköthető elemhármások szorzatösszegét. Ez ugyanis megegyezik a kifejtéssel kapott eredménnyel. A kifejtés egyébként bármelyik sor vagy oszlop szerinti is lehet.

Tehát az  $n$ -ed rendű determinánusra visszatérve, ott is sor vagy oszlop szerinti kifejtést kell alkalmaznunk, s az így kapott  $n-1$ -ed rendű aldeterminánsokat kell kiszámolnunk. Tehát  $n$ -ed rendű determináns a következő értéket értjük:

$$\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = a_{11}A_{11} + a_{12}A_{12} + \dots + a_{1n}A_{1n} = \sum_{i=1}^n a_{1i}A_{1i},$$

ahol  $A_{1i}$  az  $a_{1i}$  elemhez tartozó  $n-1$ -ed rendű aldetermináns, amelyet úgy kapunk, hogy az első sor és az  $i$ -edik oszlop elhagyásával adódó  $n-1$ -ed rendű determinánst  $(-1)^{1+i}$ -vel megszorozzuk. De akár  $n$ -ed rendű determinánusra is alkalmazható a Sarrus-szabály.

**TRANSZPONÁLT MÁTRIX FOGALMA:**

Ha az  $n*m$  típusú  $\mathbf{A}$  mátrix sorait és oszlopait egymással felcseréljük, akkor kapjuk az  $\mathbf{A}$  mátrix transzponáltját. Mely  $m*n$  típusú lesz.  $[a_{ij}]^T = [a_{ji}]$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \text{K} & a_{n1} \\ a_{12} & a_{22} & \text{K} & a_{n2} \\ & \text{M} & & \\ a_{1m} & a_{2m} & \text{K} & a_{nm} \end{bmatrix}$$

**MÁTRIX ADJUNGÁLTJA:**

Egy négyzetes mátrix adjungált mátrixának, vagy röviden adjungáltjának nevezzük az

$$\text{adj } A = \text{adj} \begin{bmatrix} a_{11} & a_{12} & \text{K} & a_{1n} \\ a_{21} & a_{22} & \text{K} & a_{2n} \\ & \text{M} & & \\ a_{n1} & a_{n2} & \text{K} & a_{nn} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{21} & \text{K} & A_{n1} \\ A_{12} & A_{22} & \text{K} & A_{n2} \\ & \text{M} & & \\ A_{1n} & A_{2n} & \text{K} & A_{nn} \end{bmatrix}$$

mátrixot, ahol  $A_{ij}$  az  $\mathbf{A}$  mátrix determinánsának  $a_{ij}$  eleméhez tartozó előjeles aldeterminánsa. Látható, hogy **adj**  $\mathbf{A}$   $i$ -edik sorának  $k$ -edik eleme az  $|\mathbf{A}|$  determináns  $k$ -edik sora  $i$ -edik eleméhez tartozó előjeles aldetermináns, vagyis **adj**  $\mathbf{A}$  az aldetermináns-mátrix transzponáltja.

**INVERZ MÁTRIX FOGALMA:**

Ha  $\mathbf{A}$  négyzetes mátrix és  $|\mathbf{A}| \neq 0$ , akkor  $\mathbf{A}$  inverz (vagy reciprok) mátrixán azt az  $\mathbf{A}^{-1}$  szimbólummal jelölt mátrixot értjük, amelyre:

$$\mathbf{A} * \mathbf{A}^{-1} = \mathbf{E}, \text{ illetve } \mathbf{A}^{-1} * \mathbf{A} = \mathbf{E}.$$

$\mathbf{E}$  az egységmátrixot jelöli. A fenti definícióból következik, hogy a jobb és baloldali inverz mátrix megegyezik, és

$$A^{-1} = \frac{\text{adj } A}{|A|}.$$

Abban az esetben ha az  $\mathbf{A}$  mátrix nem kvadratikus, de az egyenletrendszerünk megkövetelné az inverz mátrix használatát, akkor helyette az egyenletrendszert átírhatjuk kompakt mátrix alakba.

Legyen az egyenletrendszerünk

$$[y] = [N] * [x]$$

alakú, ahol a  $[x]$  elemei alkotják az ismeretlenek halmazát.

Ha  $[N]$  négyzetes mátrix, akkor  $[x]$  kifejezhető

$$[x] = [N]^{-1} * [y]$$

alakban.

Ha viszont  $[N]$  nem négyzetes mátrix, akkor a következő átalakítás hajtható végre:

$$[N]^T * [y] = [N]^T * [N] * [x]$$

$$[x] = [[N]^T * [N]]^{-1} * [N]^T * [y]$$

A programban implementálásra kerültek ezek a mátrixműveletek. Ízelítőként bemutatnám, hogyan is valósul meg a mátrixszorzás:

```
function szoroz(matrix1:TMatrix;x1,y1:integer;matrix2:TMatrix;x2,
                y2:integer): TMatrix;
var i,a,b: integer;
    sz: TMatrix;
begin
    if x1<>y2 then
        begin
            exit;
        end;
    setlength(sz,x2,y1);
    for a:=0 to y1-1 do
        begin
            for i:=0 to x2-1 do
                begin
                    sz[i,a]:=0;
                    for b:=0 to y2-1 do
                        sz[i,a]:=sz[i,a]+(matrix1[a,b]*matrix2[i,b]);
                    end;
                end;
            end;
        szoroz:=sz;
    end;
```

A `TMatrix` egy kétdimenziós, dinamikus tömb. A függvény megkapja paraméterként a két összeszorozandó mátrixot, valamint azok méreteit, s visszatérési értéként megadja a szorzás eredményeként létrejött mátrixot.

## 2.2. Ponttranszformációk

Ponttranszformáción egy olyan megfeleltetést értünk, amely során egy ponthalmaz minden egyes pontjához hozzárendelünk egy-egy pontot.

A transzformáció általános formája

$$p' = Mp,$$

ahol  $p$  a transzformálandó pont,  $p'$  a transzformált pont helyvektora,  $M$  pedig a transzformáció mátrixa.

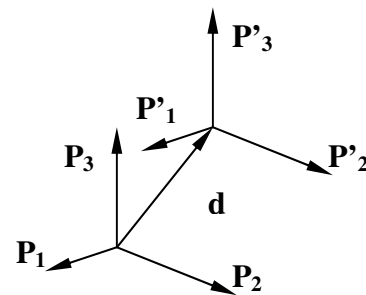
### EGYBEVÁGÓSÁGI TRANSZFORMÁCIÓK:

Egybevágósági ponttranszformáció esetén fix vonatkoztatási rendszer mellett az objektumunk helyzete változik meg, úgymond elmozdul. Ezek közül a következő kettőről szeretnék pár szót ejteni.

*Eltolás:*

A  $p$  pontot eltoljuk a  $d(d_x, d_y, d_z)$  vektorral

$$M = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

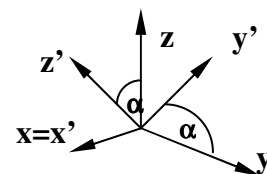


*Elforgatás:*

Egy pontot elforgathatunk adott  $\alpha$  szöggel valamelyik koordinátatengely körül. Az óramutató járásával megegyező forgatási irányhoz negatív, az ezzel ellentéteshez pedig pozitív forgatási szög tartozik.

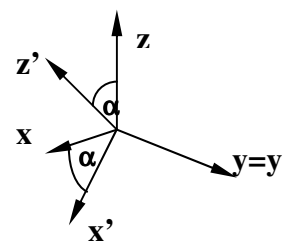
Forgatás az  $x$  tengely körül

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$



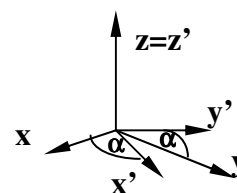
Forgatás az  $y$  tengely körül

$$M = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$



Forgatás a  $z$  tengely körül

$$M = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$



### **PROJEKTÍV TRANSZFORMÁCIÓ:**

Az ilyen leképezés is ponthoz pontot rendel, de lényeges különbség, hogy ez dimenzióvesztő transzformáció, vagyis nem kölcsönösen egyértelmű megfeleltetés. Az ilyen leképezést szokás elfajuló leképezésnek is hívni. Ezen belül a térnek síkra történő leképezéséről lesz szó. Ide tartozik például a centrális projekció (vetítés), a párhuzamos vetítés és az axonometria.

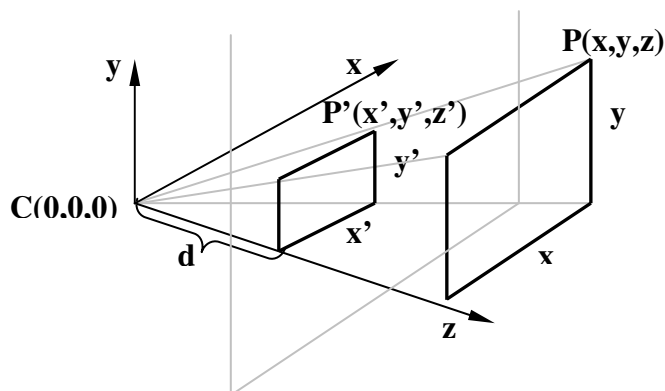
*Centrális projekció (Középpontos vetítés):*

Középpontos vetítés esetén a következőket tehetjük fel: a képsík az  $x$ - $y$  koordinátasík és a vetítési centrum a  $z$  tengely pozitív felén van. Ha ezek a feltételek nem lennének adottak, az sem nagy probléma, mert koordináta-rendszer transzformációval bármikor elérhetjük, hogy ezek a feltételek teljesüljenek. (Kivéve persze ha a vetítési centrum illeszkedik a képsíkra) Figyelnünk kell még az úgynevezett eltűnési síkra is. Ez alatt azt, a vetítési centrumon áthaladó, a képsíkkal párhuzamos síkot értjük, melynek pontjainak vetülete a képsík végtelen távoli egyenesére illeszkedik. A centrális projekció lineáris, nem párhuzamosságtartó és kettősviszonytartó leképezés. Minden térbeli pontot úgy képezünk le a síkra, hogy a térbeli pont és a  $C$  középpont által alkotott egyenessel kidőfött pont lesz a pont képe a síkon.

A gyakorlatban a képsíknak nem egy végtelen sík, hanem annak egy lehatárolt, jelentősen korlátozott része. Azért, hogy úgymond 'bent maradjunk' ezen a

korlátozott területen, vágást kell alkalmaznunk. Ez is megvalósítható a centrális projekcióval.

A centrális projekció analitikus modellje:



1. ábra Centrális projekció analitikus modellje

A  $C$  pont a vetítési centrum, a  $d$  pedig a vetítési távolságot jelenti.

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} xd & yd & d \end{pmatrix}$$

Fontos tudni, hogy ha egymás után több transzformációt hajtunk végre az objektumunkon, akkor legutoljára hajtjuk végre a vetítést!

### 2.3. Görbék

A görbeapproximáció igen nagy fejezetet ölel fel a számítógépi grafika területén. Az approximációs görbék mindegyikénél a következő a feladat: adott pontokhoz keresünk olyan görbét, amely azon pontok által van meghatározva. Ugyanakkor kritériumnak tekintjük azt is, hogy a görbe a megadott pontokat csak közelítse, s ne menjen át rajtuk (vagy legalábbis minél kevesebben menjen át). Meg kell még azt is említünk, hogy

léteznek racionális görbék is. A racionális görbéket először S. A. Coons alkalmazta a CAGD-ben (Computer Aided Geometric Design). A racionális görbék azért napjaink egyik legelterjedtebb görbeleírási módszerei, mert sok alakváltoztatási lehetőséget biztosítanak, leírhatók velük a kúpszeletek és invariánsak a projektív transzformációval szemben. A racionális görbe egyébként az eredeti görbe centrális vetülete. Ez a fejezet egy rövid áttekintést ad különböző approximációs módszerekről.

### 2.3.1. Hermite-ív:

A Hermite-ív alapvetően interpolációs eljárás. Amiért itt mégis megemlíteném, az azért van, mert a következő két alfejezetben tárgyalandó görbetípus, a Bézier-görbe és a B-spline görbe kiszámítási módja is visszavezethető a Hermite-ívnél használatos alakra speciális esetekben. A Hermite-ív definícióját a következőképpen lehet megadni:

$$s(u) = (2u^3 - 3u^2 + 1)p_0 + (-2u^3 + 3u^2)p_1 + (u^3 - 2u^2 + u)t_0 + (u^3 - u^2)t_1 \quad u \in [0,1]$$

A Hermite-ívről részletesebb leírás az 5.2 fejezetben található, az interpolációs módszerek között.

### 2.3.2. Bézier-görbe:

Ezt a módszert mondhatni egy időben, és egymástól függetlenül fejlesztette ki P. Bézier (Renault művek, 1962) és P. de Casteljau (Citroën, 1959). Görbeleírási módszerét Bézier az azóta híressé vált UNISURF CAD rendszer számára fejlesztette ki.

Adott  $p_0, p_1, \dots, p_n$   $n+1$  darab pont és legyen  $t \in [0,1]$ . Az eljárás közelíti a támpontokat, de meghatározott módon.

*Bézier-féle megvalósítás:*

Itt a támpontok súlyozott összege ad meg egy görbepontot a következő módon:

$$\underline{r}(t) = \sum_{i=0}^n p_i B_i^n(t)$$

A  $B_i^n(t)$  súlyfüggvényt *Bernstein polinom*nak szokás hívni, s kiszámítási módja a következő:

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

Bernstein polinomok tulajdonságai:

- $n$ -ed fokú polinomok a Bernstein polinomok
- $n$  fokú polinomtér bázisait alkotják
- $n$ -ed fokú tag előállítható az  $(n-1)$ -ből

$$B_i^n = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t),$$

vagyis rekurzívan előállítható

- normált bázis, vagyis

$$\sum_{j=0}^n B_j^n(t) \equiv 1$$

- minden  $B_i^n(t)$ -nek egy maximuma van, s az a  $t = \frac{i}{n}$  helyen van.

Bézier-görbe tulajdonságai:

- Globálisan változtatható a görbe  
Ha bármelyik támpontot megváltoztatom, akkor az egész görbe megváltozik.  
Vagyis minden görbepont függ az összes támponttól.
- Támpontok számának növekedésével nő a polinom fokszáma
- $(n-1)$ -szer folytonosan differenciálható
- $r(0)$  és  $r(1)$  pontok azonosak az első és az utolsó támponttal
- Az érintők az első és az utolsó pontban:

$$\underline{r}(0) = n \overrightarrow{p_1 p_0} \qquad \underline{r}(1) = n \overrightarrow{p_{n-1} p_n}$$

A kapcsolat miatt lényegesek ezek az érintők.

- Lineáris precizitás  
A Bézier-görbe képes leírni egy egyenest, ha a támpontokat is egy egyenesen vesszük fel.
- Affin invariancia  
A Bézier-görbe invariáns az affin transzformációkkal szemben, vagyis a támpontrendszer affin transzformációja ugyanazt a görbét eredményezi, mint a görbe pontonkénti affin transzformációja.  
Projektív transzformációra ez nem igaz.
- Konvex burok tulajdonság  
A Bézier-görbe nem hagyja el a konvex burkát.
- 'Variation diminishing'  
Ez a tulajdonság a konvex burkon belüli viselkedést írja le. Ha veszünk egy félegyenest, akkor a Bézier-görbének ugyanannyi, vagy kevesebb metszéspontja van a félegyenessel, mint amennyi metszéspontja van a félegyenestnek a kontrollhálózattal (kontrollpoligonnal).
- Egyenes szakasz és parabolaív modellezhető vele, de nem tudunk kört vagy ellipszist rajzolni vele
- Térgörbe is lehet, amennyiben a támpontok térben helyezkednek el

*De Casteljau-féle szerkesztési eljárás:*

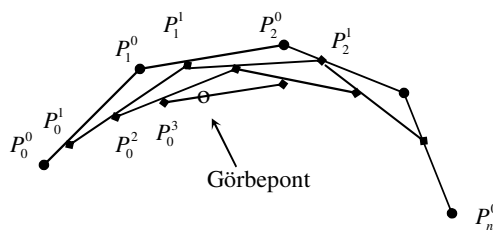
Rekurzív rajzolóalgoritmus alapulhat erre a szerkesztési módra, mely a következőképpen néz ki:

$$p_i^r(t) = (1-t)p_i^{r-1} + t p_{i+1}^{r-1}(t)$$

ahol  $r$  a rekurziós szintet jelöli és  $t \in [0,1]$ .  $r=1, \dots, n$  és  $i=0, 1, \dots, n-r$ , valamint

$$p_i^0 = p_i \quad (i = 0, 1, \dots, n).$$

Ez az eljárás kettévágja a Bézier-görbét két azonos számú kontrollponttal meghatározott görbére.



2. ábra Bézier-görbe szerkesztése de Casteljau-algoritmussal,  $t = \frac{1}{3}$  esetén

Megnézzük, hogy a két pont közötti távolság kisebb-e, mint egy előre definiált érték. Amíg nem éri el ezt az értéket, addig rekurzívan folytatjuk az algoritmust a bal oldali részgörbével.

A Bézier-görbét másképpen is bevezethetjük. Tekintsük adottnak a  $p_0, p_1, p_2, p_3$  approximálandó pontokat. Ekkor legyen

$$\begin{aligned}t_0 &= s(0) = 3(p_1 - p_0) \\t_3 &= s(1) = 3(p_3 - p_2)\end{aligned}$$

Látható, hogy így az alapfeltételek megegyeznek a Hermite-ívnél megköveteltekkel. Így a következő végeredményt kapjuk:

$$s(u) = (-u^3 + 3u^2 - 3u + 1)p_0 + (3u^3 - 6u^2 + 3u)p_1 + (-3u^3 + 3u^2)p_2 + 3u^3 p_3$$

Természetesen ez csak harmadfokú Bézier-görbékre alkalmazható. Harmadfokú Bézier-görbékét lehet úgy egymáshoz csatolni, hogy a csatlakozás folytonos legyen.

*Racionális Bézier-görbe:*

Az  $n$ -ed rendű racionális Bézier-görbét a következő kifejezéssel definiálhatjuk:

$$\underline{r}(t) = \frac{\sum_{i=0}^n w_i p_i B_i^n(t)}{\sum_{j=0}^n w_j B_j^n(t)}, \quad t \in [0,1]$$

ahol  $p_i$  ( $i=0, 1, \dots, n$ ) továbbra is a kontrollpontokat (támpontokat) jelöli, a  $B_i^n(t)$  a Bernstein polinomok, a  $w_i$  skalárokat pedig súlyoknak nevezzük. A súlyokat legtöbb esetben pozitívként szokás felvenni, különben szingularitásokat okozhatnak. A súlyok azt határozzák meg, hogy a görbe mennyire közelítsen meg egy-egy kontrollpontot.

### 2.3.3. B-Spline görbék:

A B-Spline görbékhez a következő négy nevet szokták hozzákapcsolni: Böhm, de Boor, Cox és Reisenfeld; bár mint látni fogjuk, ettől több matematikus munkássága is kiterjedt erre a területre.

Bár a B-spline görbék alapjául szolgáló B-Spline alapfüggvények már régóta ismertek (a XIX. században Lobacsevszkij vizsgálta egy speciális esetét, 1946-os cikkében pedig Schoenberg használta statisztikai adatok simítására), görbék tervezésére azonban csak a hetvenes évek elején váltak használhatóvá. Ehhez nagymértékben hozzájárult a B-Spline alapfüggvények rekurzív tulajdonságát felfedező C. de Boor, M. Cox és L. Mansfield, valamint a Bézier és a B-Spline görbék közötti kapcsolatot kimutató W. Gordon és R. Riesenfeld munkássága. Ez utóbbi felismerés tette lehetővé, hogy a korábban többnyire csak approximáció-elméleti szempontból vizsgált függvényeket paraméteres görbék előállítására, görbék tervezésére használják. A B-Spline görbék használhatóságához jelentősen hozzájárult W. Böhm is.

A spline szó eredetileg egy régi hajókészítő szerszám neve, ennek segítségével készítették a hajók ívelt részeit. Ez az eszköz egyfajta rugalmas vonalzóként képzelhető el. A matematikai illetve számítástechnikai spline feladata megegyezik a hajókészítő eszköz feladatával. Mi sem mutatja ezt jobban, minthogy spline-ok adják a hátterét azoknak a számítógépes tervezőrendszereknek, melyek segítségével manapság az autók, hajók és egyéb objektumok modelljét tervezik.

A B-Spline görbe bemutatásával majd a következő fejezet foglalkozik részletesebben, most csak azt szeretném bemutatni, hogy hasonlóan a Bézier-görbe esetéhez, hogyan néz ki az  $s(u)$  B-Spline görbe egyenlete Hermite-féle megközelítésben. A B-Spline görbe ívek összekapcsolásából jön létre. Építsük most fel a görbénket harmadfokú ívekből. Ekkor a  $Q_i$  ívet a  $p_{i-1}$ ,  $p_i$ ,  $p_{i+1}$ ,  $p_{i+2}$  pontok fogják meghatározni. Vagyis adottnak tekinthető 4 pont, ami megegyezik a Hermite-ív egyik lehetséges kiindulási feltételével. Innen már különböző átalakítások során megkapjuk, hogy a  $Q_i$  ív egyenlete a következő:

$$Q_i(u) = \frac{(-u^3 + 3u^2 - 3u + 1)}{6} p_{i-1} + \frac{(3u^3 - 6u^2 + 4)}{6} p_i + \frac{(-3u^3 + 3u^2 + 3u + 1)}{6} p_{i+1} + \frac{u^3}{6} p_{i+2}$$

Természetesen ez csak abban a speciális esetben alkalmazható, ha harmadfokú ívek összekapcsolásából jön létre a B-Spline görbénk.

A B-Spline görbének több különböző típusa létezik, van racionális változata ennek is, továbbá a tervezőrendszerek fejlődésével kialakultak a  $\beta$ - és a  $\nu$ -spline-ok.

*Racionális B-Spline:*

A racionális B-Spline görbét hasonlóan alakítjuk ki, mint a racionális Bézier görbét a Bézier-görbéből. Képletük is majdnem megegyezik, csak a Bernstein polinom helyére a B-Spline alapfüggvényt kell beírni. A racionális B-Spline definíciója tehát:

$$r(u) = \frac{\sum_i w_i p_i N_i^k(u)}{\sum_j w_j N_j^k(u)}$$

A  $w_i$  itt is súlyértéket jelöl, mely azt határozza meg, hogy a görbe mennyire közelítse meg a  $p_i$  kontrollpontot, s itt is pozitívak szoktak lenni ezek az értékek. Hogy a többi elem mit jelöl, arról a következő fejezet ad információkat.

A racionális B-Spline görbékben belül két csoportra oszthatjuk a görbét aszerint, hogy milyen értékeket vesznek fel a csomóértékek. (A csomóértékek szerepéről szintén a következő fejezetben lesz szó.) Tehát beszélhetünk egyenközű és nem egyenközű B-Spline-okról. Egyenközű (uniform) B-Spline esetén a csomóértékek egymástól egyenlő távolságra helyezkednek el, például a csomóértékek megegyeznek a természetes számok felsorolásával. A nem egyenközű (non uniform) B-Spline számít az egyik legáltalánosabb görbeleírási módszernek. Ennek a görbetípus nevének ismert rövidítése a NURBS (Non Uniform Rational B-Spline).

### 3. B-Spline görbe

Az előző fejezetben már bemutatásra került, hogy mikor is alakult ki a B-Spline görbe, s hogy több változata is létezik, de még igazából nem tudjuk, mi is alkotja ezt a görbeleírási módszert. Hát lássuk most, miről is van szó.

#### 3.1. A B-Spline görbe definíciója

Adott:

$$p_0, \dots, p_n \text{ } k \text{ és } u_0, u_1, \dots, u_{n+k+1};$$

ahol  $p_i$ -k a kontrollpontokat jelölik,  $k$  a görbe fokszáma, az  $u_i$ -k pedig a csomóértékek. A csomóértékeknél figyelniük kell arra az értékadásnál, hogy a

$$u_0 \leq \dots \leq u_k \leq \dots \leq u_n = \dots = u_{n+k+1}$$

feltételt az értékek kielégítsék. Megjegyzendő, hogy az  $u_0$  és az  $u_{n+k+1}$  nem hat a görbére.

A B-Spline görbe definíciója:

$$\underline{r}(u) = \sum_i^n p_i N_i^k(u) \quad u \in [u_k, u_{n+1}]$$

Ezt a görbét  $k$ -ad fokú vagy  $(k+1)$ -ed rendű görbének nevezzük, ahol  $N_i^k(u)$  a  $k$ -ad fokú normalizált B-Spline alapfüggvény. A  $p_i$  kontrollpontokat szokás de Boor pontoknak is nevezni, az általuk meghatározott poligont pedig kontroll- vagy de Boor-poligonnak.

### 3.2. Az alapfüggvény

A normalizált B-Spline alapfüggvényt a következő definícióval adhatjuk meg:

$$N_i^0(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}) \\ 0 & \text{egyébként} \end{cases}$$

$$N_i^r(u) = \frac{u - u_i}{u_{i+r} - u_i} N_i^{r-1}(u) + \frac{u_{i+r+1} - u}{u_{i+r+1} - u_{i+1}} N_{i+1}^{r-1}(u)$$

ahol  $r=1, \dots, k$  a rekurziós szintet jelöli, az  $u_i$  skalárok pedig a csomóértékek.

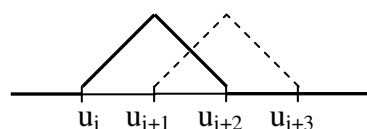
Figyelni kell az esetlegesen létrejövő  $\frac{0}{0}$  értéket, és akkor azt 0-nak kell tekintenünk. Ez az eset akkor alakulhat ki, ha a paraméterértékeknél egyenlő értékek fordulnak elő (a szomszédos csomóértékek egybeesnek).

Az alapfüggvény rendelkezik egy olyan tulajdonsággal, miszerint hiába nő a kontrollpontok száma, a Bézier-görbével ellentétben, az alapfüggvény, s ebből kifolyólag a B-Spline görbe fokszáma sem fog nőni. Nézzünk pár példát különböző fokszámú normalizált B-Spline alapfüggvényekre.

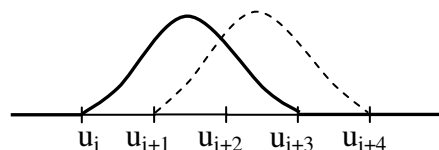
- Nulladfokú alapfüggvény,  $N_i^0(u)$



- Elsőfokú alapfüggvény,  $N_i^1(u)$



- Másodfokú alapfüggvény,  $N_i^2(u)$



Az alapfüggvényt leíró eljárás Delphi-ben a következőképpen lett megvalósítva:

```
function TForm1.Bsp_fgv(r,i:integer;u:real):real;
var szamlalo1, szamlalo2, nevezol, nevezo2, szorzol, szorzo2:real;
begin
  if (r=1) then
  begin
    if (u>=csomoertek[i]) and (u<csomoertek[i+1]) then Bsp_fgv:=1
    else Bsp_fgv:=0;
  end
  else
  begin
    szamlalo1:=u-csomoertek[i];
    nevezol:=csomoertek[i+r-1]-csomoertek[i];
    szamlalo2:=csomoertek[i+r]-u;
    nevezo2:=csomoertek[i+r]-csomoertek[i+1];
    if (nevezol=0) then
      szorzol:=0
    else
      szorzol:=szamlalo1/nevezol;
    if (nevezo2=0) then
      szorzo2:=0
    else
      szorzo2:=szamlalo2/nevezo2;
    Bsp_fgv:=szorzol*Bsp_fgv(r-1,i,u)+szorzo2*Bsp_fgv(r-1,i+1,u);
  end;
end;
```

### A B-Spline alapfüggvény tulajdonságai:

- 'local support'

$$N_i^r(u) = 0, \quad \text{ha} \quad u \notin [u_i, u_{i+r+1}]$$

- pozitív értékű függvény

$$N_i^r(u) \geq 0 \quad u \in \mathfrak{R}$$

- normált bázis, vagyis

$$\sum_{i=0}^n N_i^r(u) \equiv 1 \quad \forall u \in [u_i, u_{i+1}]$$

- az  $N_i^r(u)$  függvény  $(r-1)$ -szer differenciálható
- Bármely  $N_i^r(u)$ -nak 1 maximuma van
- Az alapfüggvényhez tartozó csomóértékek multiplicitása maximum  $r+1$  lehet

Az előző fejezetben szó volt arról, hogy megkülönböztetünk egyenközű (uniform) és nem egyenközű (non-uniform) B-Spline görbéket. Így az alapfüggvények esetén is beszélhetünk uniform változatról.

A uniform bázisfüggvény a következő alakban írható fel.

$$N_i^r(u) = \frac{u-i}{r} N_i^{r-1}(u) + \frac{i+r+1-n}{r} N_{i+1}^{r-1}(u)$$

Uniform bázisfüggvény esetén a csomóértékek megadása megegyezik a természetes számok felsorolásával, így  $u_0 = 0, \Lambda, u_i = i, \Lambda$ ; s ezért adható meg az alapfüggvény a fenti alakban.

### 3.3. A görbe tulajdonságai

Mint azt már tudjuk, egy  $k$ -ad fokú B-Spline görbe definíciója

$$r(u) = \sum_{i=0}^n p_i N_i^k(u)$$

ahol  $p_0, \Lambda, p_n \in E^3$  a kontrollpontok,  $u_0, \Lambda, u_{n+k+1}$  a csomóértékek, és  $n, k \in \mathbb{N} \quad n \geq k \geq 1$ .

Hasonlóan a Bézier-görbéhez, ezt a görbét is jellemzik speciális tulajdonságok. Felfedezhető sok hasonlóság a két görbe között, de mint majd láthatjuk, lényegi

különbségek is lesznek, melyek többnyire a B-Spline görbe rugalmasságát és jobb hasznosíthatóságát bizonyítják.

#### A B-spline görbe tulajdonságai:

- Lokális változtathatóság

Ez a tulajdonság az alapfüggvény 'local support' tulajdonságából vezethető le. Az, hogy a görbe lokálisan változtatható, azt jelenti, hogy a görbe valamely kontrollpontjának a pozícióját ha megváltoztatjuk, az nem fogja eredményezni a teljes görbe alakjának változását.

- A  $p_j$  pont hatása a  $[u_p, u_q]$  paraméterintervallumon történik meg, ahol

$$p = \max\{j, k\}$$

$$q = \min\{j + k + 1, n + 1\}$$

- Az  $u$  paraméter értékére,

$$u \in [u_j, u_{j+1}] \subset [u_k, u_{n+1}] \quad \text{a } p_{j-k}, \Lambda, p_j \text{ kontrollpontok hatnak.}$$

Ezzel igazolhatjuk lokális változtathatóságot.

- Affin invariancia

A kontrollpont-rendszer affin transzformációja ugyanazt a görbét eredményezi, mint a görbe pontonkénti affin transzformációja.

- Erős konvex burkok tulajdonság

$$\underline{r}(u) \quad u \in [u_j, u_{j+1})$$

A görbe a  $p_{j-k}, \Lambda, p_j$  kontrollpontok konvex burkában mozog. Ha a figyelembe veendő csomóértékek között van egynél nagyobb multiplicitású, akkor a számításba veendő kontrollpontok száma csökken. Megfigyelhetjük azt is, hogy a fokszám növekedésével a görbe egyre jobban távolodik a kontrollpoligontól, illetve  $k=1$  esetén a 'görbe' egybeesik a kontrollpoligonnal.

- Csomópontok multiplicitása

Egy csomóérték multiplicitása  $m$ , ha az egymás után következő  $m$  db csomóérték értéke megegyezik.

Ha  $u_{j+1} = \Lambda = u_{j+k} = \bar{u}$ , akkor  $p_j = \underline{r}(\bar{u})$ , vagyis a görbe belefut a  $p_j$  pontba, s így törés keletkezik a görbében.

- Egy támpont  $k$ -szoros megismétlésének eredménye

Ha  $k$  db támpontot ugyanazon pozícióban veszem fel, vagyis  $p_j = \Lambda = p_{j+k-1}$ ,

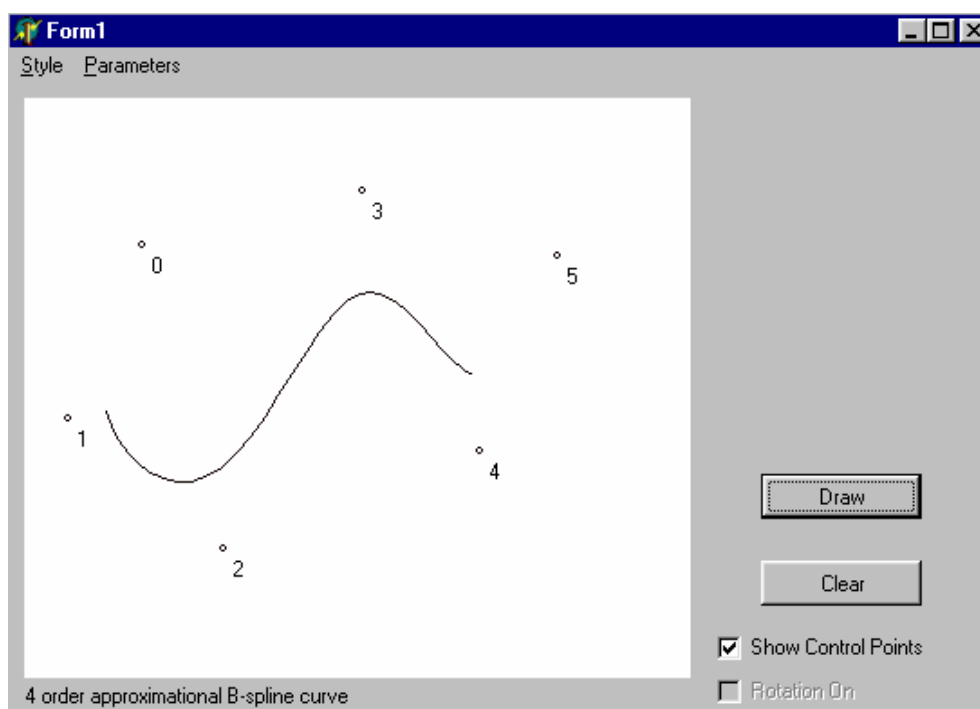
akkor a  $p_j$  pont görbepont lesz.

- Zárt görbe konstruálható vele

$p_j = p_j \bmod n + 1$ , tehát átfedés kell, hogy legyen a kontrollpontoknál.

## 4. Approximáció

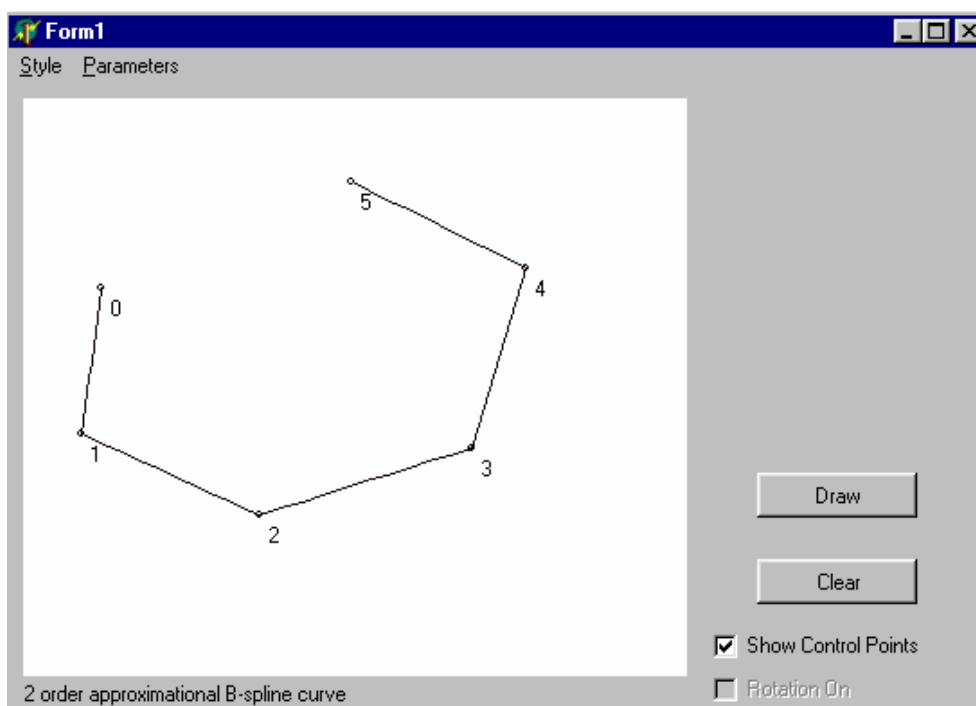
Ebben a fejezetben azt szeretném bemutatni, hogy hogyan is valósítottam meg az approximációs eljárást, valamint igazolni szeretném a görbe néhány fontos és látványos tulajdonságát. De nézzük először, hogy milyen görbét is alkot a program.



3. ábra Klasszikus, 4-ed rendű B-Spline görbe

### 4.1. Elsőfokú görbe szerkesztése

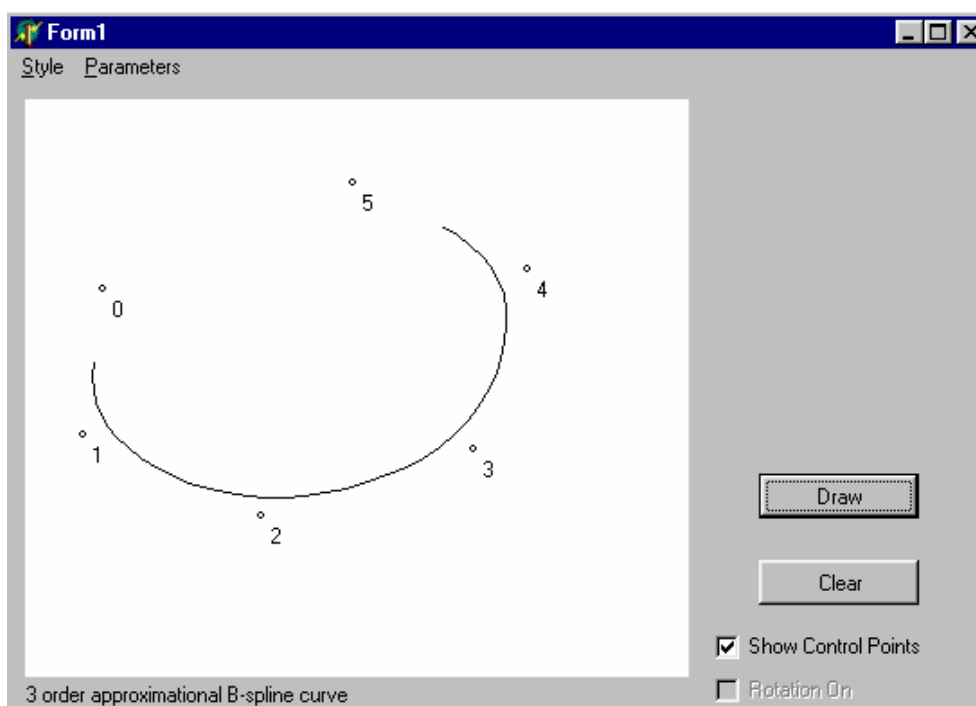
Mint azt már említettem, a görbe fokszáma illetve rendje meghatározza a görbe alakját. Minél nagyobb a fokszám, a görbe annál inkább távolodik a kontrollpoligontól.



4. ábra Elsőfokú B-Spline görbe

Abban a speciális esetben, ha a görbe fokszáma 1, akkor magát a kontrollpoligont kapjuk, mint azt a fenti képen láthatjuk is.

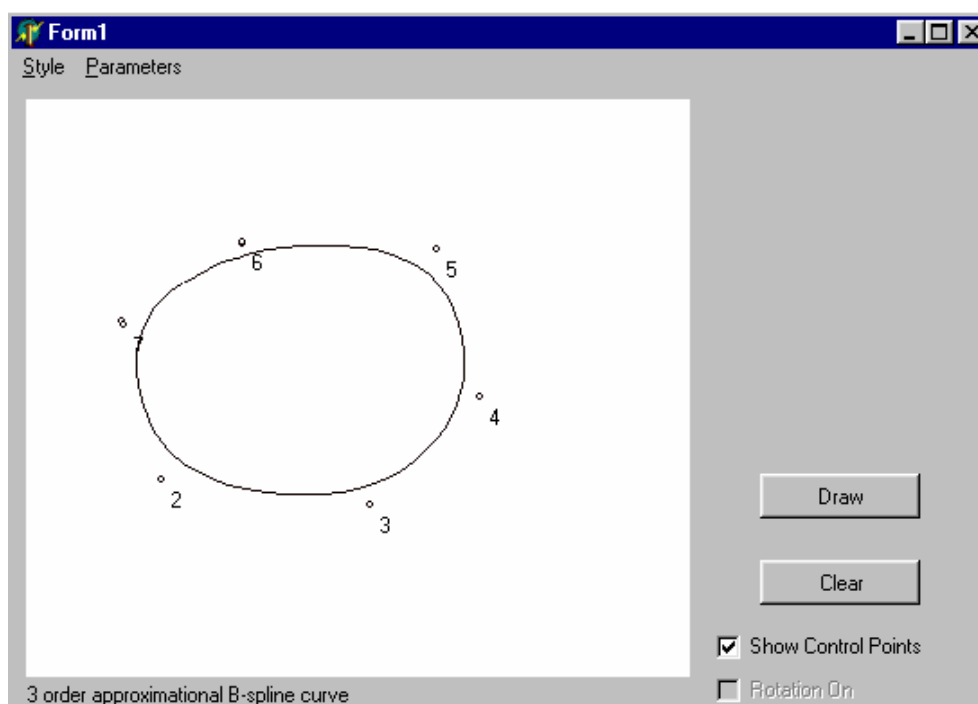
Ugyanezen kontrollpontok és csomóértékek esetén, ha a görbe rendjét 3-ra változtatjuk, akkor a következő eredményt kapjuk.



5. ábra 3-ad rendű B-Spline görbe

## 4.2. Zárt görbe szerkesztése

Ellentétben a Bézier módszerrel megbeszéltekkel, a B-spline görbe segítségével konstruálhatunk olyan görbét, amely akár kör vagy ellipszis is lehet.

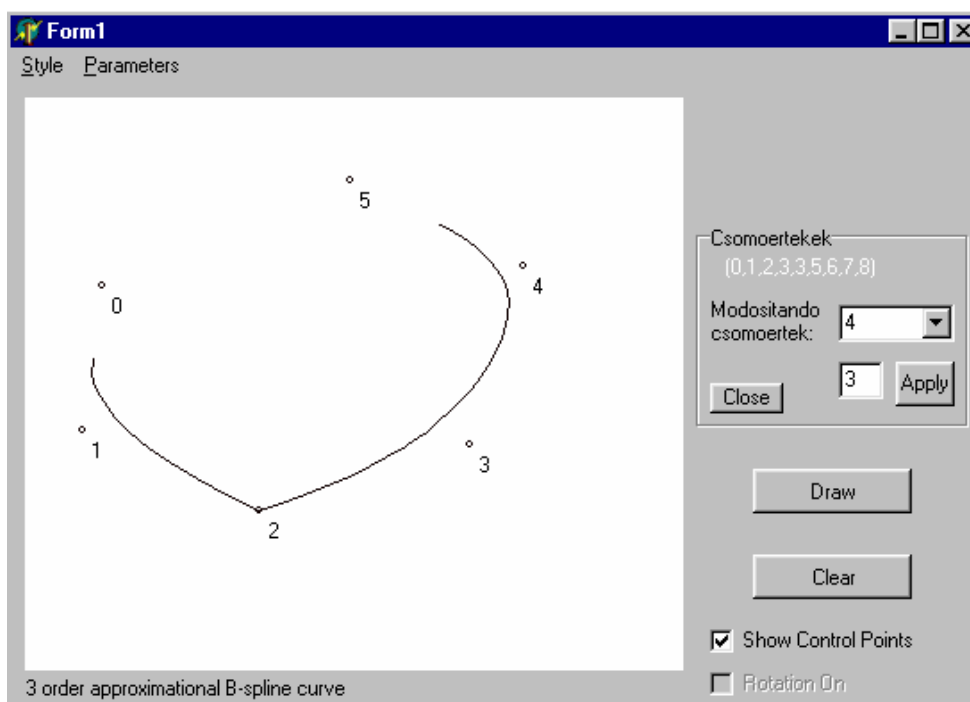


6. ábra B-Spline görbe segítségével létrehozott zárt görbe

Zárt görbét a kontrollpontok átfedésével érhetünk el. A görbe fokszáma határozza meg, hogy hány pontnak kell fednie egymást. ( $p_j = p_{j \bmod n + 1}$ .)

## 4.3. Csomóérték multiplicitása

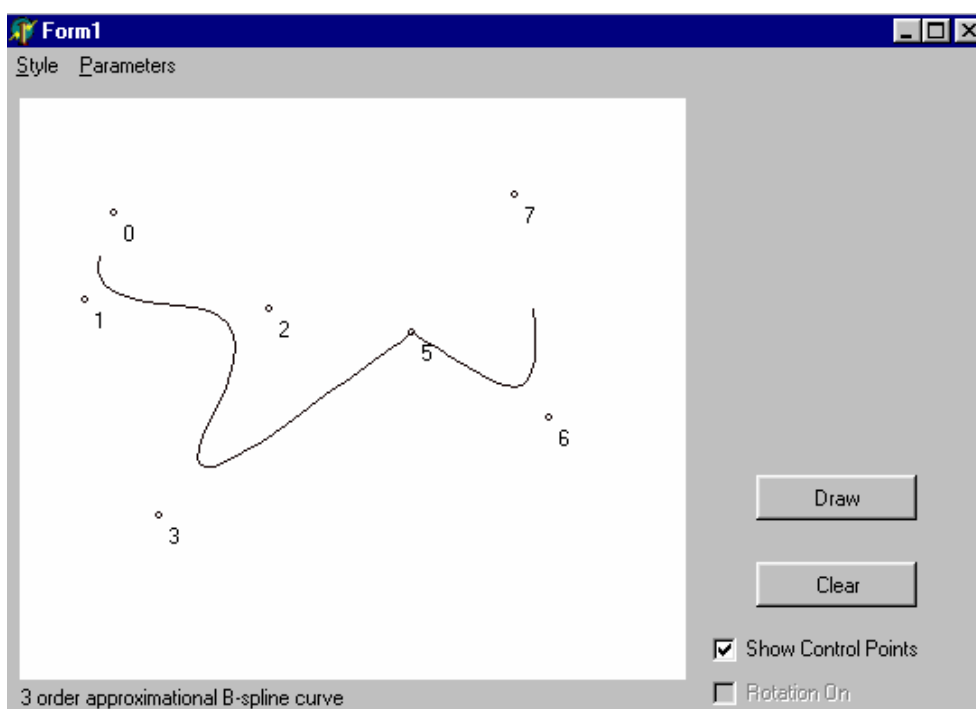
A csomóértékek igen fontos szerepet játszanak, mint paraméterértékek, a B-Spline görbék esetében. A csomóértékek variálásával megváltoztatható a görbe alakja. Mint azt majd látni fogjuk, lényeges a szerepük az interpoláció során is. Ha úgy nézzük, a csomóérték multiplicitását tekinthetjük az interpoláció egy speciális esetének is, hiszen eredményeképpen, mint azt a 7. ábra mutatja, a görbe át fog haladni egy meghatározott kontrollponton.



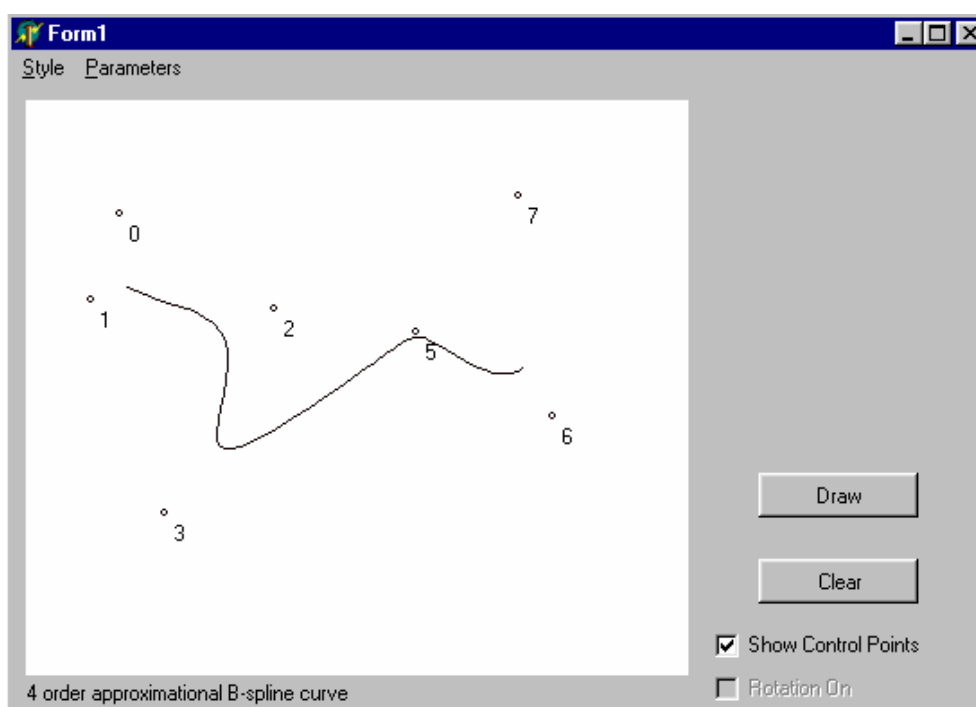
7. ábra Csomóérték multiplicitásának hatása

#### 4.4. Kontrollpontok multiplicitása

Kontrollpontok multiplicitása alatt azt értjük, hogy egynél több egymást követő kontrollpont is ugyanazokkal a koordináta-értékekkel rendelkezik, tehát egy és ugyanazon koordinátapontban találhatók. Ha  $m$  darab kontrollpontot is ugyanazon a helyen veszünk fel, és  $m$  megegyezik a görbe fokszámával, akkor ezzel ugyanolyan hatást érünk el, mintha a megfelelő csomóértékeket többszöröztük volna. Vagyis a görbénk bele fog futni az egy helyen felvett kontrollpontok halmazába. Viszont ha  $m$ -nél kevesebbszer ismétlem meg az adott alappontot, akkor már az így kapott görbe nem fog illeszkedni a pontra, hanem csak nagyobb mértékben fogja azt megközelíteni. Ez a tulajdonság könnyen szemléltethető azáltal, hogy megismétlünk egy kontrollpontot  $m$ -szer, ahol  $m$  megegyezik a görbe fokszámával, majd megnöveljük a görbe fokát. Így az is jól láthatóvá válik majd, hogy a görbe rendjének emelkedésével, a kapott görbe egyre inkább távolodik a kontrollpoligonjától.



8. ábra  $p_4=p_5$ , fokszám=2



9. ábra  $p_4=p_5$ , fokszám=3

## 5. Interpoláció

A B-Spline görbét alapvetően approximációs görbének tekintjük, ugyanakkor használható interpolációs eljárásként is. Interpoláció esetén a célunk az, hogy az eljárás során eredményül kapott görbénk átmenjen a kontrollpontokon. A kontrollpontok interpolációját megoldhatjuk úgy, ha visszavezetjük az approximációs feladatok megoldására. Első lépésként tekintsük úgy, hogy a megadott kontrollpontok a majdani görbe pontjai. Egyéb ismert paraméterek segítségével felírhatunk egy egyenletrendszert, ahol a kontrollpontok a görbepontoknak felelnek meg, s ismeretlennek tekintjük azokat az új kontrollpontokat, amelyekre ha alkalmazzuk az approximációs eljárást, egy olyan görbét kapunk, melynek pontjai az eredeti kontrollpontok. Vagyis az újonnan kiszámolt kontrollpontokra alkalmazzuk az approximációs eljárást a szokott módon.

### 5.1. Interpolációs eljárások

Több különféle interpolációs technika is ismert, melyek különbözőképpen valósítják meg az interpolációt, ebből következően a görbe alakja is eltérő, módszerenként más. Ez abból adódik, hogy egy görbe alakját a kontrollpontjain kívül többfajta paraméter is befolyásolja. Ilyen paraméter lehet például az érintővektor. Meg kell jegyezni azonban azt is, hogy esetenként más és más információkkal rendelkezünk egy görbéről, tehát nem mindig ismert számunkra az összes paraméter. Így az, hogy milyen interpolációs technikát választunk, attól is függ, hogy milyen paraméterek állnak a rendelkezésünkre, mit tudunk a görbéről.

Látható, hogy egy interpolációs feladatnak rengeteg megoldása van, s minden megoldás más és más szempontból tekinthető optimálisnak. Ugyanakkor talán

kijelenthetjük, hogy tökéletes, minden igényt maradéktalanul kielégítő megoldás nem létezik. Ebből is látható, hogy már a módszer kiválasztása is milyen lényeges momentum. A következőkben szó lesz néhány interpolációs módszerről, melyek mindegyike alkalmazható mind síkban és térben.

### 5.1.1. Lagrange interpoláció

A Lagrange interpoláció jól ismert módszer a numerikus matematika területén, ahol is függvényinterpolálásra használják. Alapelve a következő.

Legyenek adottak a  $p_i$  pontok, és  $u_i$  paraméterértékek. ( $i = 0, 1, \dots, n$ , és a paraméterértékek egymástól különbözőek.) Keresünk egy olyan  $p(u)$  legfeljebb  $n$ -ed fokú polinomot, amelyre teljesül, hogy:

$$p(u_i) = p_i \quad (i = 0, 1, \dots, n)$$

A feladat egyik megoldása lehet a

$$p(u) = \sum_{i=0}^n p_i L_i^n(u) \quad ,$$

ahol  $L_i^n(u)$ -t Lagrange-féle interpolációs alappolinomnak hívjuk, és

$$L_i^n(u) = \frac{\prod_{\substack{j=0 \\ j \neq i}}^n (u - u_j)}{\prod_{\substack{j=0 \\ j \neq i}}^n (u_i - u_j)} \quad , \text{ vagyis}$$

$$L_i^n(u) = \frac{(u - u_0) \Lambda(u - u_{i-1})(u - u_{i+1}) \Lambda(u - u_n)}{(u_i - u_0) \Lambda(u_i - u_{i-1})(u_i - u_{i+1}) \Lambda(u_i - u_n)}$$

Ebből a definícióból következik, hogy  $\sum_{i=0}^n L_i^n(u) = 1$  és  $L_i^n(u_j) = \delta_{ij}$ , ahol  $\delta_j$  a Kronecker-delta.

Ezt a legfeljebb  $n$ -ed fokú polinomot máshogyan is megkaphatjuk. Legyenek ugyanazok adottak, mint az előzőekben. Ezúttal az interpoláló polinom legyen

$$p(u) = \sum_{j=0}^n a_j u^j$$

alakú. A feltételből, miszerint  $p(u_i) = p_i$  -ből felírható egy lineáris egyenletrendszer  $a_i$  ismeretlenekre, mely mátrix alakban a következőképpen néz ki:

$$\begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} 1 & u_0 & u_0^2 & \Lambda & u_0^n \\ 1 & u_1 & u_1^2 & \Lambda & u_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & u_n & u_n^2 & \Lambda & u_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix},$$

röviden  $p = aV$ . A  $V$ -t a az interpoláció Vandermonde féle determinánsának nevezzük.

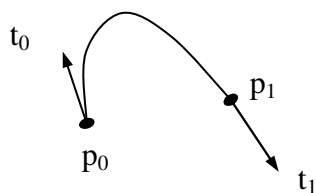
Mindkét esetben egy  $n$ -ed fokú interpolációs polinommal dolgozunk. Az ilyen  $n$ -ed fokú polinomoknak van egy kellemetlen tulajdonsága, melyet úgy hívnak, hogy oszcillálás. Ez azt jelenti, hogy a kapott interpolációs görbén olyan kiugrásokat tapasztalunk, amit az adott pontok nem indokolnának. Ezért van az, hogy a Lagrange-féle interpolációs technikát nem igazán használják görbeinterpolációra. Ugyanakkor mégis fontos szerepe van ennek a módszernek, mert sok interpolációs eljárás polinommal írja le a görbét. Ugyanakkor nem egy globális,  $n$ -ed fokú görbét használnak erre a célra, hanem több, alacsonyabb rendű polinommal meghatározott ívekből állítják elő a kívánt interpolációs görbét.

### 5.1.2. Hermite-ív

Hermite-interpoláció esetén 3 különböző esetet különböztethetünk meg. Adottnak tekinthető két pont és a hozzájuk tartozó érintővektorok, három pont és egy érintő, vagy négy pont.

Nézzük most azt az esetet, amikor már nemcsak az interpolálandó pontokat tekintjük ismertnek, hanem az érintővektorokat is.

Legyen adott a  $p_0$  és  $p_1$  pont, valamint a  $t_0$  és  $t_1$  érintővektorok.



Keresünk egy olyan

$$a(u) = a_0 + a_1 u + a_2 u^2 + a_3 u^3, \quad u \in [0,1]$$

harmadfokú polinommal meghatározott görbét, amelyre

$a(0) = p_0, a(1) = p_1, \dot{a}(0) = t_0, \dot{a}(1) = t_1$  teljesül. Ezek alapján a következő polinom együtthatókat kapjuk

$$\begin{aligned} a_0 &= p_0, \\ a_1 &= t_0, \\ a_2 &= 3(p_1 - p_0) - 2t_0 - t_1, \\ a_3 &= -2(p_1 - p_0) + t_0 + t_1 \end{aligned}$$

Ha behelyettesítjük ezeket az együtthatókat az  $a(u)$  egyenletbe, akkor átrendezés után ezt az alakot kapjuk:

$$a(u) = p_0(2u^3 - 3u^2 + 1) + p_1(-2u^3 + 3u^2) + t_0(u^3 - 2u^2 + u) + t_1(u^3 - u^2), \quad u \in [0,1]$$

Ebből felírható a görbe mátrix reprezentációja:

$$a(u) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ t_0 \\ t_1 \end{bmatrix}$$

Vezessük be a következő jelölést:

$$\begin{aligned} H_0^3(u) &= 2u^3 - 3u^2 + 1, \\ H_1^3(u) &= -2u^3 + 3u^2, \\ H_2^3(u) &= u^3 - 2u^2 + u, \\ H_3^3(u) &= u^3 - u^2 \end{aligned}$$

Így az egyenletünk alakja leegyszerűsödik, s a következő alakban írható fel:

$$a(u) = \begin{bmatrix} H_0^3(u) & H_1^3(u) & H_2^3(u) & H_3^3(u) \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ t_0 \\ t_1 \end{bmatrix}$$

A  $H_i^3(u)$  ( $i = 0,1,2,3$ ) függvényeket harmadfokú Hermite-polinomoknak nevezzük. Mint azt már említettük, ennél a módszernél nemcsak az interpolálandó pontok ismertek, hanem az érintővektorok is. Fontos tulajdonsága ennek a módszernek, hogy az

érintővektor iránya befolyásolja a görbe alakját. De nemcsak az iránya, hanem a kezdő- és végpontbeli érintő hossza is befolyásolja az ív alakját. Ez a tulajdonság sokat segít a görbe tervezésénél, hiszen tapasztalatok alapján tudni fogjuk, hogyan vegyük fel az érintővektorokat, ahhoz, hogy a kívánt alakú görbét kapjuk eredményül.

A harmadfokú interpoláció már igen népszerűnek és széles körben elterjedt eljárásnak számít, hiszen ez az a legalacsonyabb fokszám, amellyel térgörbe előállítható.

### 5.1.3. Interpolációs spline-ok

Arról már volt szó a korábbiakban, hogy előszeretettel alkalmazzák a polinomokat interpolációs görbék előállítására. Ez arra vezethető vissza, hogy a polinomok könnyen kezelhetők, gyorsan kiértékelhetők és számos más kedvező tulajdonsággal rendelkeznek. Viszont ellentétben a Lagrange-féle interpolációs technikával, nem egy magas fokszámú polinomot használnak erre a célra, hanem több, alacsonyabb fokszámú polinomot. Ez annyit jelent, hogy az interpolációs görbét több alacsonyabb rendű (többnyire harmad rendű), egymáshoz kapcsolódó görbeívekből állítják elő. S mint azt már korábban szintén említettük, az ilyen ívekből összeállított görbét spline-nak nevezzük.

Mivel a görbe ívekből áll, ezért igen fontos szerepet játszik az ívek kapcsolódása. Nézzük meg, mit is jelent az, hogy a kapcsolódás folytonos. Ez definíció szerint így hangzik:

A  $c_{i-1}(u)$  és  $c_i(u)$  legalább  $n$ -szer folytonosan differenciálható ívek az  $u=u_i$ -nél  $n$ -ed rendben folytonosan kapcsolódnak egymáshoz ( $n$ -ed rendben érintkeznek), ha

$$\frac{d^r}{du^r} c_{i-1}(u_i) = \frac{d^r}{du^r} c_i(u_i) \quad (r = 0, 1, \dots, n)$$

Ezt röviden  $C^n$  folytonosságnak szokás nevezni.

Ebből a definícióból adódik, hogy elsőrendű folytonosság esetén az íveknek nemcsak közös pontjuk van,  $u_i$ -nél hanem itt az érintővektoruk is megegyezik. Tekintsünk egy olyan feladatot, melynél adottak a  $p_0, p_1, \dots, p_n$  pontok, a hozzájuk tartozó  $u_0, u_1, \dots, u_n$  paraméterértékek és a  $t_0, t_1, \dots, t_n$  érintővektorok, valamint olyan harmadrendű, első rendben folytonosan kapcsolódó görbeívekből szeretnénk előállítani, melyre teljesül, hogy

$$c(u_i) = p_i, \quad \dot{c}(u_i) = t_i \quad (i = 0, 1, \dots, n)$$

Ez a feladat könnyen megoldható harmadrendű Hermite-interpoláció ismételt alkalmazásával. Ugyanakkor elképzelhető az is, hogy nem áll rendelkezésünkre az összes kiindulási adat, amit az előbbieken feltételeztünk. Ilyenkor sok esetben visszavezethetjük a feladatot az előzőre.

Vegyünk például, hogy csak a  $p_0, p_1, \dots, p_n$  pontok és a hozzájuk tartozó  $e_0, e_1, \dots, e_n$  érintőirányok adottak, és keresünk hozzá egy olyan görbét, melyre igaz a  $C^1$  folytonosság, a görbe átmegy az adott pontokon és ott az érintő iránya a megadott irány. Ekkor feltehetjük, hogy  $|e_i| = 1$ . Annak érdekében, hogy ezt a feladatot vissza tudjuk vezetni az előzőre, meg kell határozni az  $u_i$  paraméterértékeket és a  $t_i$  érintővektorokat. A  $t_i$  érintővektor előállítását  $t_i = \lambda_i e_i$  formában történik, ahol a  $\lambda_i$  skalár az érintővektorok hosszát adja meg. A  $\lambda_i$ -nek bármilyen értéket adhatunk, elterjedt megoldás azonban, hogy az érintő hossza arányos a pontból kiinduló valamelyik húrral, mint például  $\lambda_i = \alpha |p_{i+1} - p_i|$ , ahol  $\alpha > 0$  tetszőleges skalár. Az  $u_i$  paraméterértékek meghatározására szintén több módszer is ismert. A legegyszerűbb talán az uniform (egyenlő osztásközű) paraméterezés, amikor  $u_i = i$ . Egy másik módszer a húrhosszal arányos paraméterezés. Ekkor  $u_i$  értékére teljesül a következő:

$$\frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} = \frac{|p_{i+1} - p_i|}{|p_{i+2} - p_{i+1}|}$$

Használatos még paraméterezésre a centripetális módszer is, amelynél a következő összefüggés teljesül:

$$\frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} = \sqrt{\frac{|p_{i+1} - p_i|}{|p_{i+2} - p_{i+1}|}}$$

Az uniform paraméterezés esetén a számítások leegyszerűsödnek, az interpolálandó pontok invariánsak az affin transzformációval szemben, ellentétben a többi paraméterezéssel. Ugyanakkor ezzel a módszerrel akkor kaphatunk igazán 'jó' görbét, ha a szomszédos pontok távolsága közel egyenlő. A húrhosszal arányos paraméterezés már figyelembe veszi a szomszédos pontok közötti távolságot, akár csak a centripetális módszer. Ha akarjuk a paraméterértékeket akár tetszőlegesen is megválaszthatjuk, akár csak az érintővektorok hosszát, de ne felejtsük el, hogy ezek az értékek hatással vannak a görbe alakjára, s a paraméterértékek módosításával is befolyásoljuk a görbe alakját.

Nézzünk egy másik példát is, amit vissza szeretnénk vezetni a fejezet első példájára. Tegyük fel, hogy most, csak a  $p_0, p_1, \dots, p_n$  pontok és a hozzájuk tartozó,

egymástól különböző  $u_0, u_1, \dots, u_n$  paraméterek adottak. Valamint egy olyan első rendben folytonosan kapcsolódó harmadrendű ívekből álló görbét keresünk, amely az  $u_i$  paraméternél a  $p_i$  ponton megy át. Ahhoz, hogy ezt visszavezethessük a kezdeti problémára, meg kell határoznunk a pontokhoz tartozó érintővektorokat vagy az érintőirányokat (mint láttuk ezekből meghatározhatók az érintővektorok). Az érintővektorok megadhatók például az úgynevezett Bessel-érintőkkel. Ez alatt a  $p_{i-1}, p_i, p_{i+1}$  és az  $u_{i-1}, u_i, u_{i+1}$  adatokkal meghatározott  $c_i$  parabolának az  $u_i$  helyen vett  $t_i = \dot{c}(u_i)$  érintőjét értjük.

Most tekintsünk egy másik problémakört. A kiindulási feltételek egyezzenek meg az előbbivel, tehát legyenek adottak a  $p_0, p_1, \dots, p_n$  pontok és a hozzájuk tartozó egymástól különböző  $u_0, u_1, \dots, u_n$  paraméterértékek. De most egy olyan másodrendben folytonosan kapcsolódó harmadrendű ívekből álló  $r(u)$ ,  $u \in [u_0, u_n]$  görbét keresünk, amelyre  $r(u_i) = p_i$  ( $i=0, 1, \dots, n$ ) teljesül. Megoldásként használható az úgynevezett Ferguson-spline módszer. Ez a módszer egymáshoz másodrendben folytonosan kapcsolódó harmadrendű Hermite-ívekből állítja elő a görbét. Ehhez ugye először meg kell határozni az adott pontokat, hiszen az  $r_i(u)$ ,  $u \in [u_i, u_{i+1}]$  ív a  $p_i, p_{i+1}, t_i, t_{i+1}, u_i, u_{i+1}$  adatokból állíthatók elő. A folytonossági feltételek alapján a  $t_i$ -ket meghatározhatjuk  $i = 1, \dots, n-1$ -re. Ehhez csak az kell, hogy  $\ddot{r}_{i-1}(u_i) = \ddot{r}_i(u_i)$  teljesüljön. A végpontbeli érintők meghatározására többféle módszer létezik. Ilyen például a természetes, a kvadratikus, a harmadrendű folytonosság, a Bessel, a parabola érintő és a rögzített peremfeltételek alkalmazása.

*Természetes:* az a feltevésünk, hogy a kezdő és a végpontban a görbület nulla

$$\left( \ddot{r}_0(u_0) = 0, \quad \ddot{r}_{n-1}(u_n) = 0 \right).$$

$$\text{Kvadratikus: } \ddot{r}_0(u_0) = \ddot{r}_0(u_1) \quad \text{és} \quad \ddot{r}_{n-1}(u_{n-1}) = \ddot{r}_{n-1}(u_n)$$

*Harmadrendű folytonosság:* Ebben az esetben feltételezzük, hogy az  $u_1$  és  $u_{n-1}$  pontokban harmadrendben folytonosan kapcsolódnak az ívek, vagyis a  $p_1$  és  $p_{n-1}$  pontok nem csomópontok.  $\left( \ddot{r}_0(u_1) = \ddot{r}_1(u_1) \quad \text{és} \quad \ddot{r}_{n-2}(u_{n-1}) = \ddot{r}_{n-1}(u_{n-1}) \right).$

*Rögzített:* Ekkor a  $t_0$  és a  $t_n$  érintőket tetszőlegesen megadjuk.

## 5.2. Alkalmazott interpolációs módszer

Bár a B-Spline approximációra tervezett görberajzoló eljárás, interpolációra is használható. A feladat tehát az, hogy keressünk egy olyan adott fokszámú B-Spline görbét, amely az adott paraméterértékeknél a megadott pontokon áthalad. Vagyis a feladat nem más, minthogy határozzuk meg a görbe kontrollpontjait és csomóértékeit, úgy, hogy ezt a feltételt a keletkezett görbe kielégítse. Jelöljük a görbénket  $r(u)$ -val, az interpolálandó pontokat pedig  $p_i$ -vel. Vegyük most a  $p_i$  pontot, és a hozzá tartozó  $t_i$  paramétert, ahol  $t_i \in [u_i, u_{i+1}]$ . Ekkor, ha  $p_i$  pont ( $i = 0, \dots, n$ ) illeszkedik a  $t_i$  paraméternél egy  $k$ -ad fokú

nem racionális B-spline görbére  $\left( r(u) = \sum_j d_j N_j^k(u) \right)$ , az következik, hogy kielégíti a következő egyenletrendszer:

$$p_i = r(t_i) = \sum_{j=0}^l d_j N_j^k(t_i)$$

Ez az egyenletrendszer  $n+1$  egyenletből és  $l$  ismeretlenből áll, valamint igaz, hogy  $2 \leq k \leq l \leq n$ . Továbbá a  $d_j$  kontrollpontok lesznek az egyenletrendszer ismeretlenjei. Mint látszik, az hogy mennyi kontrollpontot határozunk meg az interpolálandó pontokhoz, az - bizonyos megszorítások között - rajtunk múlik. Én az egyszerűség, és könnyebben számolhatóság kedvéért úgy döntöttem, hogy az interpolálandó pontok száma és az újonnan meghatározandó kontrollpontok, vagy más néven de Boor pontok száma egyezzen meg, tehát  $l=n$ . Az egyenletrendszernek van egy együttható mátrixa, s természetesen annyi megoldáshalmaza, ahány koordinátával rendelkezik a kontrollpont (és az interpolálandó pontok). Tehát térgörbe esetén 3. Ha kifejtjük ezt az egyenletrendszert, akkor a következő alakot kapjuk, felhasználva az előbbi kijelentést, miszerint legyen  $l=n$ :

$$\begin{aligned} r(t_0) &= N_0^k(t_0)d_0 + N_1^k(t_0)d_1 + \Lambda + N_n^k(t_0)d_n \\ r(t_1) &= N_0^k(t_1)d_0 + N_1^k(t_1)d_1 + \Lambda + N_n^k(t_1)d_n \\ &\quad \text{M} \qquad \qquad \qquad \text{M} \\ r(t_n) &= N_0^k(t_n)d_0 + N_1^k(t_n)d_1 + \Lambda + N_n^k(t_n)d_n \end{aligned}$$

Így az egyenletrendszerünk most  $n+1$  ismeretlenből és  $n+1$  egyenletből áll. Ez is felírható szokásos mátrix alakban:

$$[p] = [N][d]$$

Mint látjuk, az ismeretlenjeink úgymond rossz oldalon szerepelnek. Mivel  $n=l$ , ezért  $[N]$  kvadratikusan mátrix, így ez átírható a következő alakba:

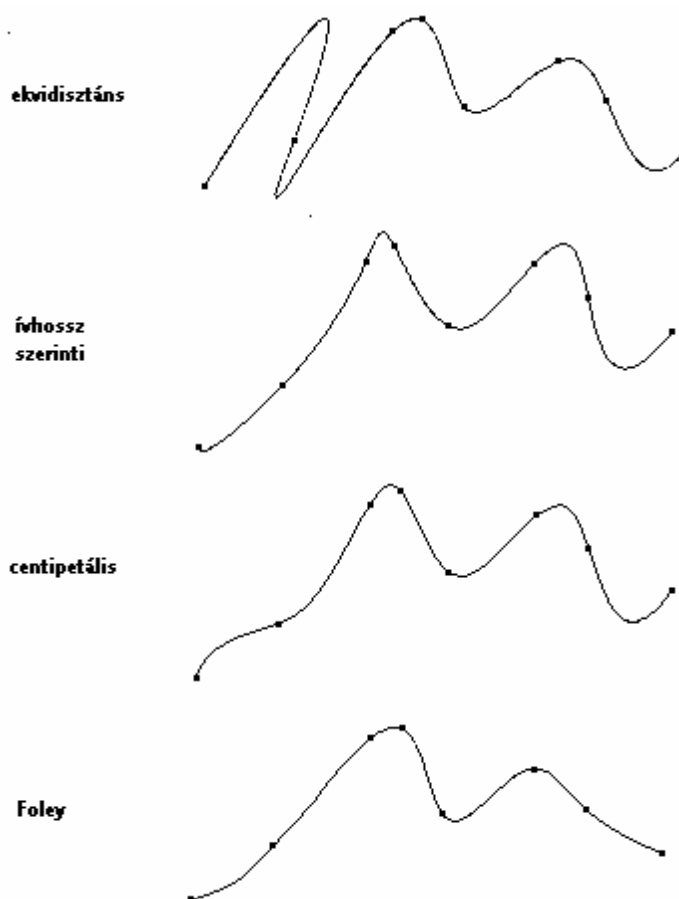
$$[d] = [N]^{-1} [p]$$

Az inverz mátrix számítására vonatkozó információkról szó volt a 2.1 fejezetben. Vagyis abban az esetben, ha  $[N]$  mégsem lenne négyzetes mátrix, akkor az egyenletrendszer át kell írni kompakt alakban, s akkor már értelmezhető lesz rajta az inverz fogalma. Az eredmény így a következő lesz:

$$\begin{aligned} [p] &= [N][d] \\ [N]^T [p] &= [N]^T [N][d] \\ [d] &= \left[ [N]^T [N] \right]^{-1} [N]^T [p] \end{aligned}$$

A programban a mátrixszámításokat egy külön unit valósítja meg.

Így már a  $d_j$  ismeretlenek kiszámíthatóak lennének, ha a  $t_i$  paramétereket ismernénk. Ezek szerint legelső lépésként ezeket kellene meghatározni. Erre is több különböző módszer áll a rendelkezésünkre, ilyen például az ekvidisztáns (egyenlőközű), húrhossz szerinti (ív hossz szerinti), centripetális és a Foley-módszer. Hogy mennyire függ a paraméterezéstől a görbe alakja, azt a következő ábra jól szemlélteti:



10. ábra Adott pontsorozat interpolálása különböző paraméterezési technikák mellett

A program az ívhossz szerinti paraméterezési módszert valósítja meg, még hozzá a következőképpen:

```

procedure t_param2(c:GPoints3D;pontszam:integer);
var i,j: integer;
    l,tav_vekt: real;
begin
    l:=0;
    for j:=1 to pontszam-1 do
    begin
        tav_vekt:=sqrt (sqr (c[j].x-c[j-1].x)+sqr (c[j].y-c[j-1].y)+
            sqr (c[j].z-c[j-1].z));
        l:=l+tav_vekt;
    end;
    for i:=0 to pontszam-1 do
    begin

```

```

if i=0 then t[i]:=0 else
if i=pontszám-1 then t[i]:=1
else
begin
t[i]:=t[i-1]+ (sqrt(sqr(c[i].x-c[i-1].x)+
sqr(c[i].y-c[i-1].y)+ sqr(c[i].z-c[i-1].z)))/l;
end;
end;
for i:=0 to pontszám-1 do
t[i]:=t[i]*csomoertek[pontszám];
end;

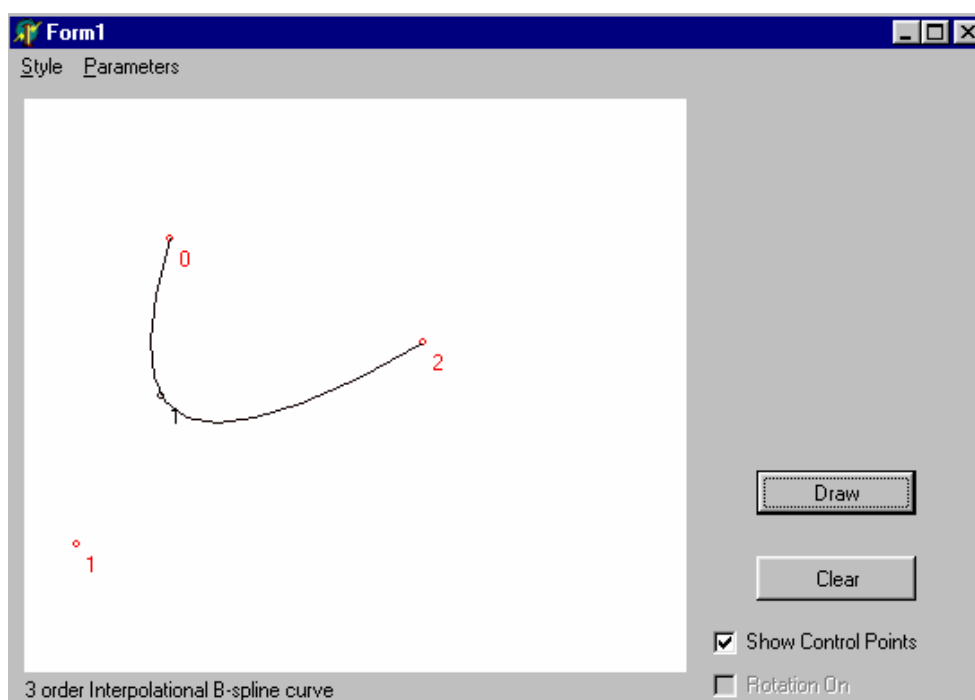
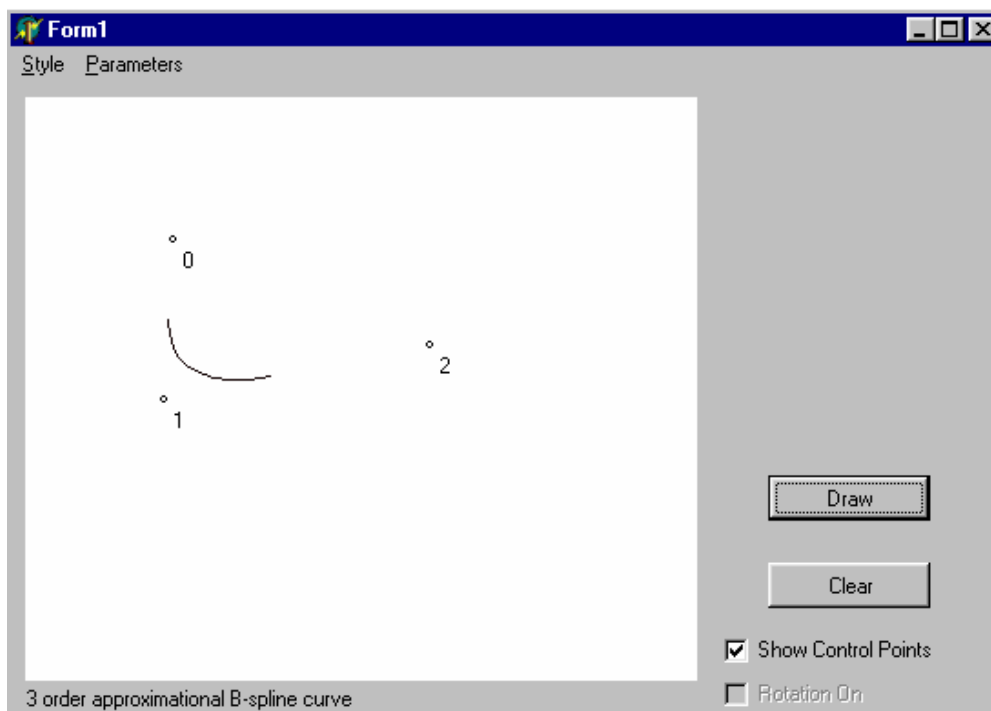
```

Arra, hogy a  $t[i]$  értéket beszorozzuk a  $csomoertek[pontszám]$  értékkel, azért van szükség, hogy biztosítsuk a  $t_i$  paraméter  $[u_i, u_{i+1}]$  intervallumba esését. Ez azért szükséges, mert adottnak tekintem az  $u_i$  csomóértékeket, így figyelni kell rá, hogy a  $t_i$  paraméter eleget tegyen az elején tárgyalt feltételnek, és értéke beleessen az intervallumba. Ugyanakkor, ha a csomóértékek nem lennének előre megadva, akkor megoldás lehetne az, hogy a csomóértékeket a  $t_i$  paraméterértékek segítségével határozzuk meg. Ez pedig a következő módon történhetne például:

$$u_0 = K = u_k = 0, \quad u_{m-k} = K = u_m = 1$$

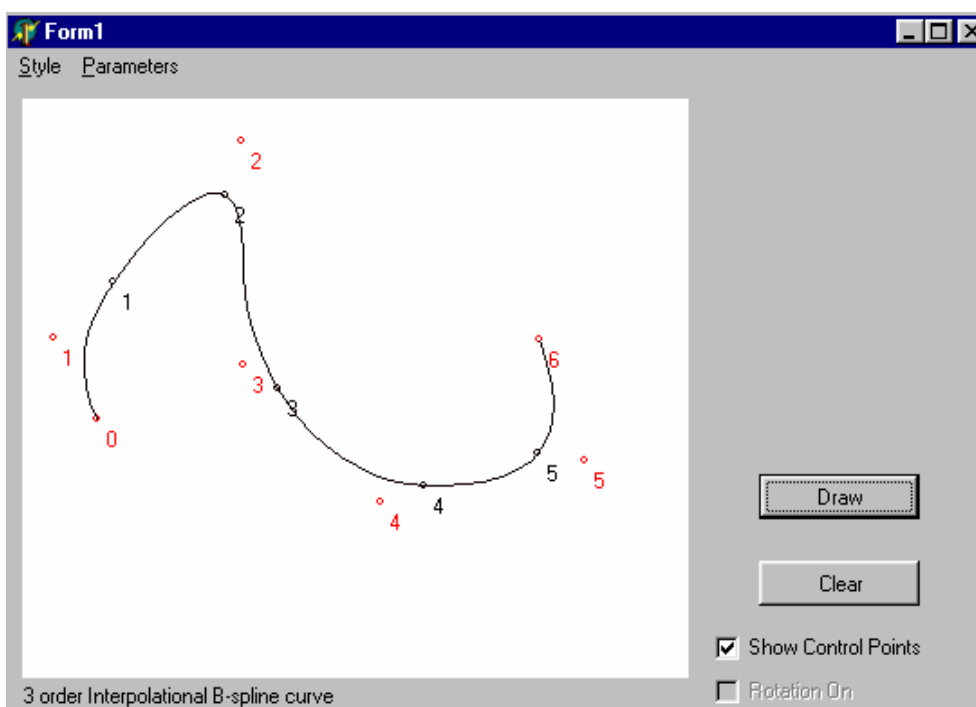
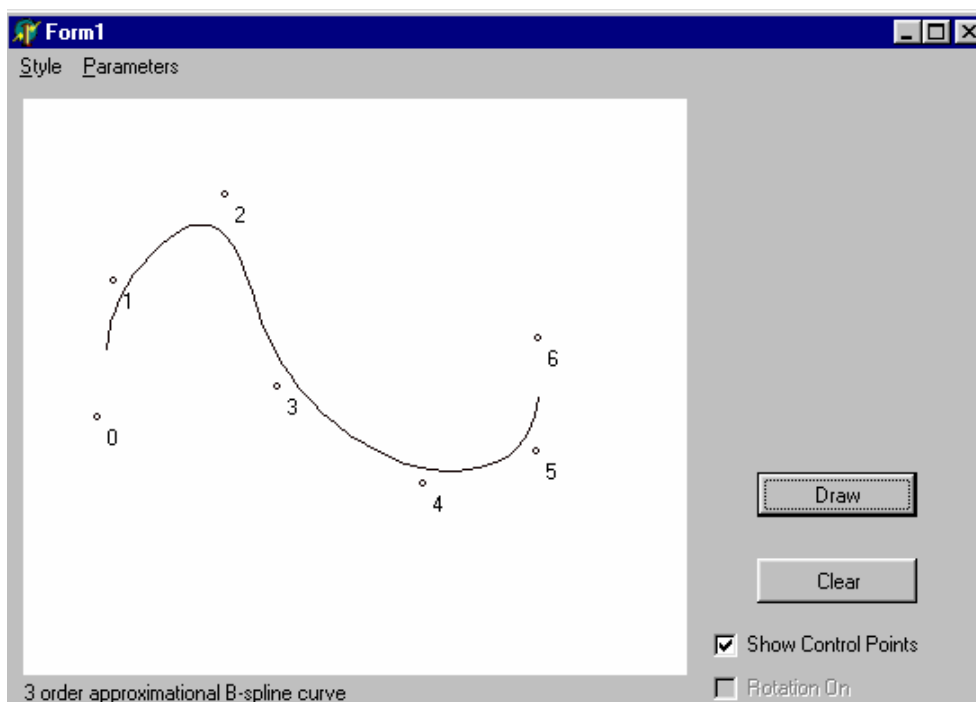
$$u_{j+k} = \frac{1}{k} \sum_{i=j}^{j+k-1} t_i, \quad j = 1, K, n-k$$

Most, hogy már kiszámoltuk a  $t_i$  paraméterértékeket, segítségükkel megoldhatjuk az egyenletrendszerünket. Így megkapjuk a  $d_j$  kontrollpontok koordinátáit. Ezeket a pontokat felhasználva most már eredeti funkciójá szerint használjuk a B-Spline görbeszerkesztési eljárást, és ezekre a kontrollpontokra végrehajtjuk az approximációt. Ez a része az algoritmusnak teljes mértékben megegyezik a hagyományos approximációval. Tehát az interpolációs probléma megoldását visszavezettük approximációs jellegű feladat megoldására. Nézzük milyen eredménnyel:



**11-12 ábra Approximáció és interpoláció azonos kiindulásból**

A két képen jól látszik, hogy ha az adott pontok mindkét esetben ugyanazok, akkor hogy néz ki a görbe approximáció, illetve interpoláció esetén, valamint jól látszik, hogy hol helyezkednek el az újonnan kiszámolt kontrollpontok az utóbbi esetben. Több pont esetén az eredmény a következő:



13-14. ábra Újabb összehasonlítás

Az előző fejezetben több interpolációs eljárás is bemutatásra került. Hogy végül is miért ez a módszer lett alkalmazva a programozás során, azt a következő fejezet fogja

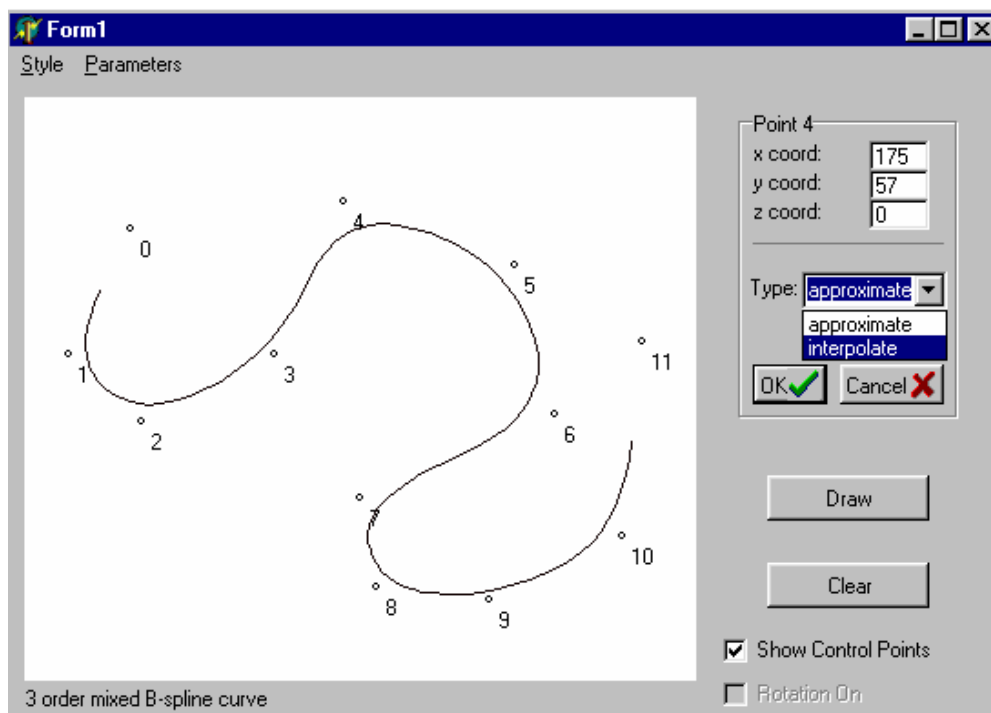
indokolni. A cél ugyanis az volt, hogy egy olyan eljárás szülessen, mely a két technika együttes használatát teszi lehetővé, s erre ez a módszer bizonyult a legideálisabbnak.

## 6. A két technika ötvözése

Arról már szót ejtettünk, miként is tudjuk megszerkeszteni egy görbe alakját különböző feltételek mellett, valamilyen igényt kielégítve. Azt is tudjuk, hogy két különböző jellegű görbét különböztethetünk meg. Vannak approximációs és interpolációs jellegű görbék, s ezek eddig mindvégig külön voltak említve. Ugyanakkor könnyen elképzelhető az az igény, hogy egy adott görbére nézve a kontrollpontok egy részén menjen át a görbénk, míg a többi kontrollpontot pedig csak közelítse meg.

Az érvek azért szóltak az alkalmazott interpolációs eljárás mellett, mert ez biztosít számunkra egy igen fontos tulajdonságot, mégpedig azt, hogy a keletkezett görbe továbbra is egy egységként lesz kezelhető. Más interpolációs technika alkalmazása esetén megeshet, hogy a görbénket interpolációs és approximációs részgörbékre kell osztani. Ekkor figyelni kellene a kapcsolódás folytonosságára, valamint egyszerre több görbét kellene 'kézben tartani' és kezelni.

Tekintsük azt az alaphelyzetet, hogy a B-Spline görbét, amit eredendően approximációs célokra terveztek, azt mi tényleg arra is szeretnénk használni. Ugyanakkor engedjük meg, hogy minden egyes kontrollpont esetén, ha akarjuk, akkor megváltoztathassuk a jellegét approximációról interpolációsra.



15. ábra Egy pont jellegének megváltoztatása

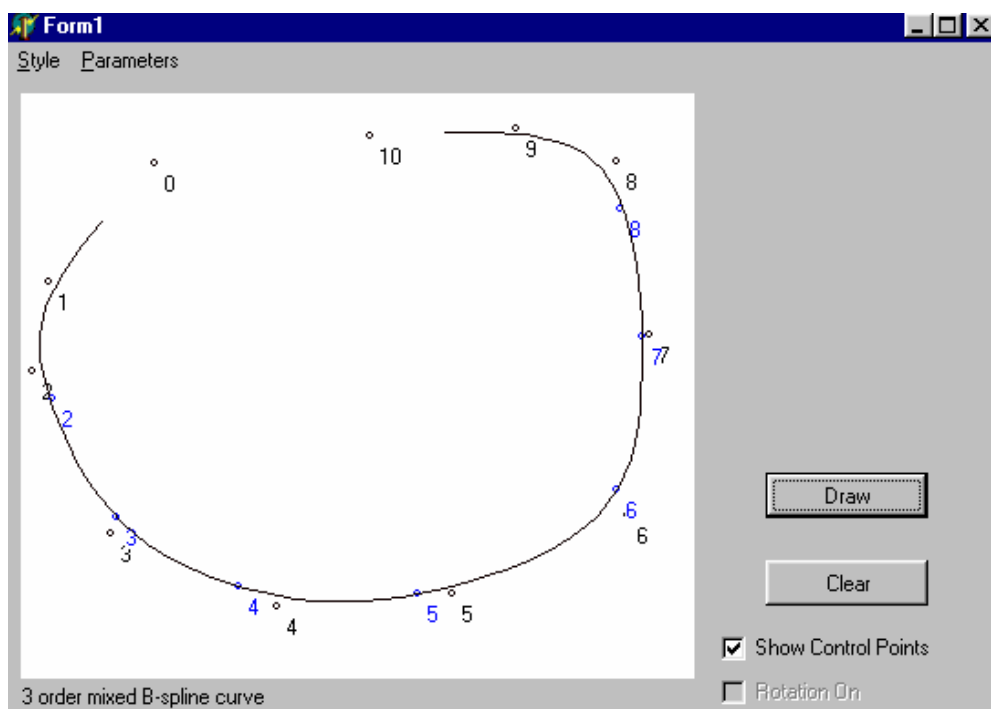
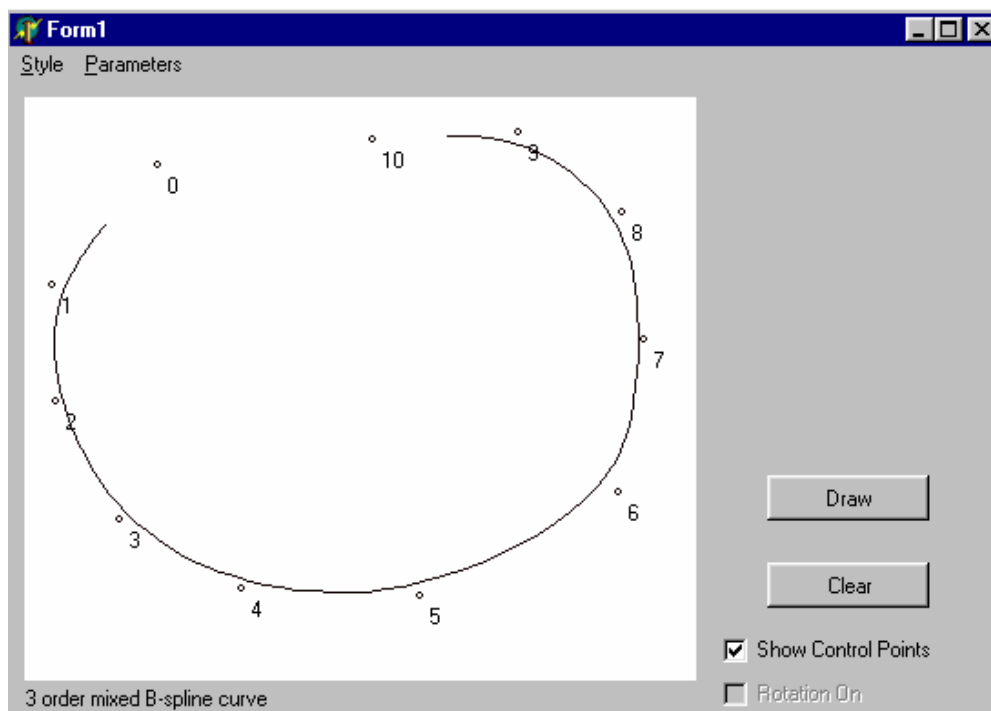
A megvalósított eljárás során először ugyan itt is részekre osztjuk a görbét, de a végén egy egészként fogjuk kezelni. A részekre bontás a következőképpen fog történni:

```
begin
  b:=false;
  index:=-1;
  for i:=0 to pontok_szama-1 do
  begin
    if (Control_Points[i].ptype=1) and (b=false) then
    begin
      inc(index);
      part[index].eleje:=i;
      b:=true;
    end;
    if (Control_Points[i].ptype=0) and (b=true) then
    begin
      part[index].vege:=i-1;
      b:= false;
    end;
  end;
end;
```

A  $\text{part}$  nevezetű tömb elemei olyan elempárosok, melyek azt adják meg, hogy egy adott interpolációs szakasz melyik pontnál kezdődik, s melyik pontig tart. Azért szükséges ezt az információt egy tömbben tárolni, mivel nem tudhatjuk, hogy egymáshoz viszonyítva mely pontok jellege kerül megváltoztatásra, így azt sem tudhatjuk hány darab részgörbénk fog keletkezni. Tehát azokat az interpolálandó pontokat tekintjük egy halmazba tartozónak, melyek egymás szomszédjai. Minden egyes halmazra meghívjuk az alkalmazott interpolációs eljárást. Erről már tudjuk, hogyan működik. Tehát kapunk egy új kontrollpont halmazt eredményül, melyeket ha approximálni fogunk, akkor az interpolálandó pontok a görbe pontjai lesznek. Így hát a következő lépésünk nem lesz más, mint hogy azokat a kontrollpontokat, melyeket interpolálandó pontoknak jelöltünk meg, azokat kicseréljük az interpolációs eljárás során kapott kontrollpontokkal. Aztán az így módosított kontrollpont-sorozatra már alkalmazhatjuk az eredeti approximációs eljárást. Mivel továbbra is egy összefüggő kontrollpont-sorozatunk van, s most már az összes görbepont jellege is megegyezik, így egy, minden igényt kielégítő görbe keletkezik végül. Látható, hogy így teljesíteni tudjuk a feltételt, s a görbe nem esik szét interpoláló és approximáló részgörbékre, hanem egy egészként tudjuk kezelni.

Mivel nem kell összekapcsolni több különböző görbeszegmenst, így nem kell a kapcsolódás folytonosságának fenntartásával sem bajlódni, és nem kell egyszerre több görbét kezelni egy adott pillanatban. Ahhoz hogy a görbénk szép és törésmentes legyen több szegmens összekapcsolása esetén  $C^2$ -folytonosságra lenne szükség.

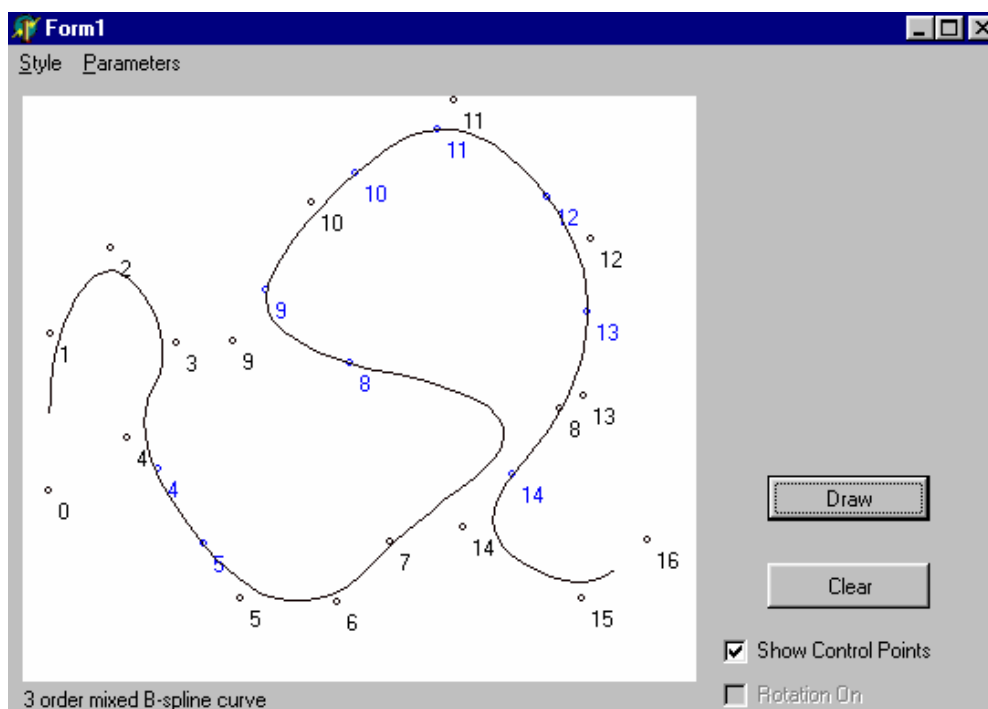
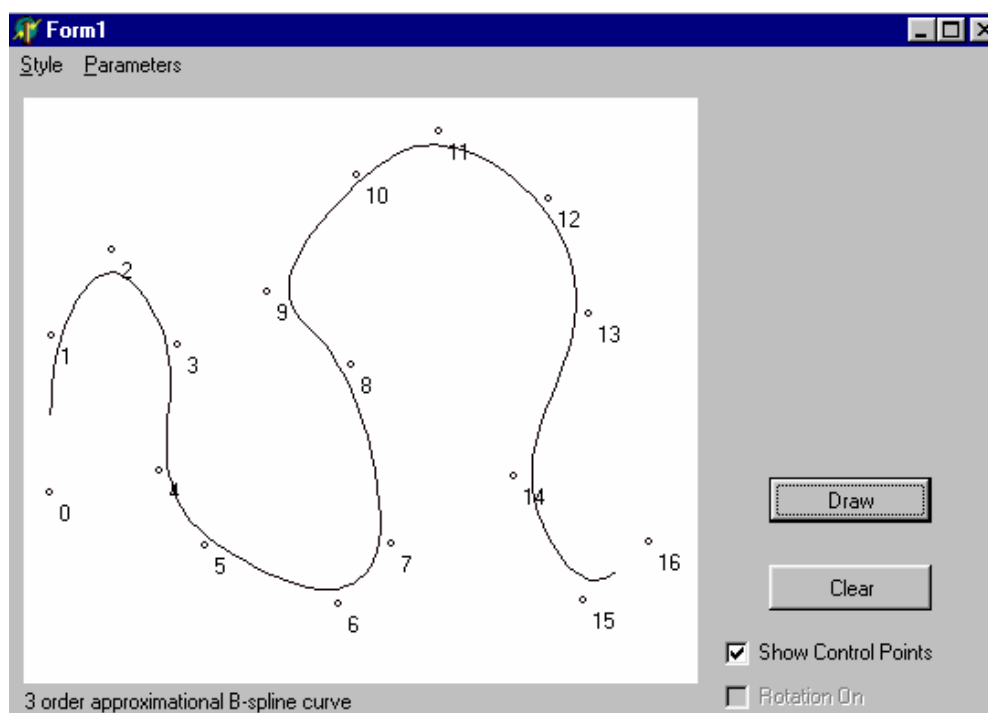
Végül az ötvözeses technika megvalósításának eredményét szintén képekben szeretném bemutatni. A képpárok első tagja az adott kontrollpontokra végrehajtott approximációs eljárás eredményeként született görbét mutatja be. A párok második tagja pedig egyes pontok interpolációs pontokká tétele után kapott görbét szemlélteti.



16-17. ábra Interpoláció és approximáció ötvözése

Az interpolációra kiválasztott pontok a következők voltak: 2,3,4,5,6,7,8. Ez tehát azt jelenti, hogy ezekre a pontokra meghívtuk az interpolációs eljárást, és az eredményül kapott kontrollpontokat beírtuk az eredeti kontrollpontok helyére. Az eredeti

kontrollpontok azért vannak feltüntetve a második ábrán is, hogy jól látható legyen az interpoláció eredményessége.



18-19. ábra Több interpolációs rész egy görbén belül

Itt látható, hogy az adott görbén belül található két interpolálandó ponthalmaz is. Az első halmaz két elemből áll, a 4-es és az 5-ös pontokból. A második halmazt pedig a 8,9,10,11,12,13,14 pontsorozat alkotja. Az eljárás ugyanazt az elvet követi, mint az előbbieken. Kigyűjtésre kerülnek azok a ponthalmazok melyek interpolációt igényelnek, majd az alkalmazott interpolációs eljárás eredményeként kapott új kontrollpontok segítségével megszerkesztjük a végleges, approximációs jellegű görbét.

Természetesen itt is igaz, hogy a paraméterértékek megváltoztatásával tovább finomítható a görbe alakja.

**IRODALOMJEGYZÉK:**

- [1] David F. Rogers, J. Alan Adams – Mathematical Elements for Computer Graphics; McGraw-Hill, New York, 1990, 2. kiadás
- [2] Josef Hoschek, Dieter Lasser – Fundamentals of Computer Aided Geometric Design; A K Peters, Wellesley, Massachusetts, 1993
- [3] Scharnitzky Viktor – Mátrixszámítás; Műszaki Kiadó, Budapest, 1996, 5. kiadás
- [4] Juhász Imre – Számítógépi geometria és grafika; Miskolci Egyetemi Kiadó, 1995
- [5] Sulyok Péter – előadásjegyzet
- [6] Hoffmann Miklós – Approximációs és interpolációs lehetőségek a számítógépi grafikában (egyetemi doktori értekezés); Debrecen, 1992
- [7] Kovács Emőd – előadásjegyzet