KURV- OCH YTAPPROXIMATION MED POLYNOM

Magnus Bondesson
Institutionen för Datavetenskap
1999-02-04, 2001-02-01 (red), 2003-02-05 (red)

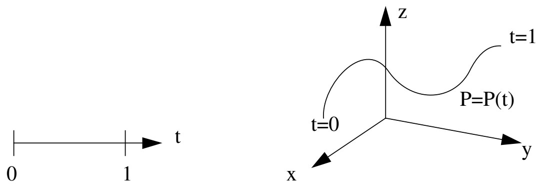
1 Allmänt om kurvapproximation med polynom

Detta papper ersätter framställningen i HB: 315-354, FvD: 468-516, Angel: 484-515 eller Hill: 597-663 för den som så önskar det. HB gör mycket bara ytterst lokalt. FvD är mer omfattande och har andra index (numrering) än jag. Angel är riktigt bra. Möller tar bara upp Bezier-material. Framför allt B-splines presenterar jag på ett förhoppningsvis aptitligare sätt.

Givet ett antal ordnade punkter
$\mathbf{P}_i = (x_i,y_i)$, i = 0,1,2,...,n
i xy-planet eller
$\mathbf{P}_i = (x_i,y_i,z_i)$
i 3D, vill man ibland sammanbinda dem med en mjuk kurva. Men det finns också fall då man nöjer sig med att den resulterande kurvan bara går i närheten av punkterna. I CAD-sammanhang där kurvorna modelleras fram är det viktigt att en flyttning av en punkt bara påverkar kurvan lokalt, dvs i närheten av punkten som flyttas. Tillämpningsområden inom datorgrafik är modellering och animering.

Det enklaste vore att bara sammanbinda punkterna med räta linjer, men oftast vill man att kurvan skall ha högre regularitet än så. Man eftersträvar åtminstone kontinuerlig lutning och helst även kontinuerlig krökning, vilket ställer krav på kontinuerliga första och andra derivator.

Den resulterande kurvan kommer att beskrivas på parameterform
$\mathbf{P}=\mathbf{P}(t)$, $0 \le t \le n$ (eller ett lägre tal i vissa fall)
oberoende av om vi befinner oss i 2D eller 3D. Första punkten på kurvan motsvarar $\mathbf{P}(0)$ och sista $\mathbf{P}(n)$. Vi avbildar sålunda ett intervall på en kurva, se följande figur.



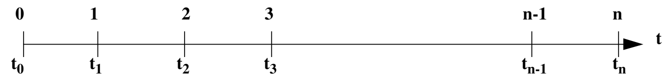
Exempelvis kan en rät linje mellan två punkter $\mathbf{P}_0$ och $\mathbf{P}_1$ skrivas på formen $\mathbf{P}=\mathbf{P}(t)=\mathbf{P}_0+t\mathbf{S}$, $0 \le t \le 1$, där riktningsvektorn $\mathbf{S} = \mathbf{P}_1-\mathbf{P}_0$. En och samma kurva kan beskrivas med många olika parameterframställningar. Från rent geometrisk synpunkt är det nästan (jfr avsnitt om G-kontinuitet i FvD) likgiltigt vilken vi väljer. Om något skall röra sig längs kurvan är det lämpligt att parametern t är proportionell mot den tillryggalagda kurvlängden. Parametern kan då fysikaliskt tolkas som tiden eller kurvlängden. Det är ju däremot lättare sagt än gjort när man bara utgår från ett antal punkter.

2 Global interpolation med polynom

Man vill använda enkla funktioner för att beskriva kurvorna. Inget är enklare än polynom, vilka också används genomgående. Även om det torde vara välbekant för alla, påminner jag om att global interpolation med polynom inte är så bra. Den resulterande kurvan uppvisar ofta mycket kraftiga oscillationer. I det fallet beskrivs kurvan med ett polynom (för resp koordinat) med högt gradtal. Man kan skriva

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} L_i(t)\boldsymbol{P}_i$$

där basfunktionen eller blandningsfunktionen (eng blending function) $L_i(t)$ är ett polynom av gradtalet n och talar om hur stort inflytande punkten $\mathbf{P}_i$ har för parametervärdet t. Vi inför parametervärden t_i motsvarande punkterna $\mathbf{P}_i$, dvs $\mathbf{P}(t_i) = \mathbf{P}_i$. Dessa kan t ex väljas som i figuren nedan. De behöver inte vara ekvidistant belägna.

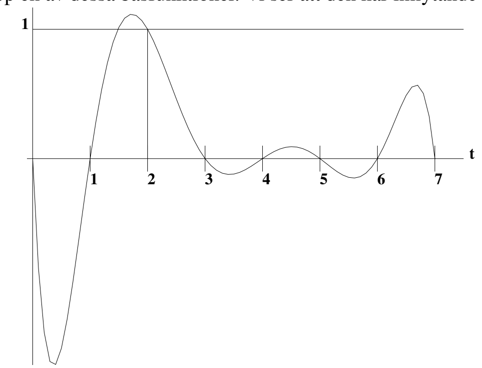


Figur 1. Speciella parametervärden.

L_i konstrueras så att $\mathbf{P}_i$ har fullt inflytande för parametervärdet t_i motsvarande $\mathbf{P}_i$, men inte något för övriga givna punkter, dvs

$$L_i(t) = \begin{cases} 1, & t = t_i \\ 0, & t = t_j,\ j \neq i \end{cases}$$

Nedan har vi ritat upp en av dessa basfunktioner. Vi ser att den har inflytande över hela intervallet.



Figur 2. Basfunktionen $L_2(t)$ för ett fall med 8 punkter. Vi ser att funktionen inte nödvändigtvis har maximalvärdet 1. (MB: SPLINEPAK.c).

Formeln för basfunktionerna är

$$L_i(t) = \prod_{j=0,\ j \neq i}^{n} \frac{t - t_j}{t_i - t_j}$$

3 Lokal interpolation med polynom

I stället skall vi använda styckevis polynomapproximation, dvs den resulterande kurvan kommer att bestå av ett antal hopfogade polynomkurvor. Vi gör detta genom att för varje enskilt parameterintervall $[t_i, t_{i+1}]$ approximera med ett polynom, t ex i 2D och med tredjegradspolynom

$$\boldsymbol{P}_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} = \boldsymbol{a}_0 + \boldsymbol{a}_1 t + \boldsymbol{a}_2 t^2 + \boldsymbol{a}_3 t^3$$

där $\mathbf{a}_j$ är vektorer med två komponenter och dessutom är olika från polynom till polynom (dvs beror på intervallnumret, vilket vi dock ej sätter ut). Lämpligen ser man på x och y var för sig. Alla kurvor av denna typ, dvs hopsatta av polynomkurvor, kallas för **splines**.

Vi kommer här huvudsakligen att se på fallet att de lokala kurvorna är tredjegradskurvor (i den meningen att deras parameterframställningar är av gradtalet 3). För jämförelsens skull tas ibland även gradtalen 1 och 2 upp. Vissa böcker ger en betydligt allmännare framställning.

Ett antal metoder beskrivs i följande avsnitt. Vi går från det enklaste till det som idag är "state-of-the-art" i CAD-program, nämligen NURBS-kurvor. Inget av stegen på vägen är bortkastat. Vi går däremot inte alls in på dataanpassningsmetoder, som t ex minsta-kvadrat-metoden.

4 Hermiteinterpolation

Den här metoden har inget direkt praktiskt intresse men är enkel och utgör grund för en senare. De lokala tredjegradskurvorna skall ha de givna punkterna som ändpunkter, dvs

$$(1) \qquad \begin{cases} \boldsymbol{P}_i(t_i) = \boldsymbol{P}_i \\ \boldsymbol{P}_i(t_{i+1}) = \boldsymbol{P}_{i+1} \end{cases}$$

för i=0,1,...,n-1. Detta ger 2 av de 4 villkor som behövs för att bestämma $\mathbf{a}_0$,$\mathbf{a}_1$,$\mathbf{a}_2$ och $\mathbf{a}_3$. Man arbetar härvid med fördel i en lokal parameter u, varvid u=0 motsvarar t_i och u=1 motsvarar t_{i+1}.

I Hermites metod förutsättes även att derivatorna med avseende på t är kända i de olika punkterna och är (om 2D) $\mathbf{P}'_i = (x'_i(t_i), y'_i(t_i))$.

Notera emellertid att även om vi anser oss känna lutningen hos kurvan i de givna punkterna så är dessa derivator inte kända. I 2D gäller ju för kurvlutningen att $\frac{dy}{dx} = \frac{y'(t)}{x'(t)}$, vilket innebär att x'_i och y'_i bara kan bestämmas på en konstant faktor när.

Vi tar nu tilläggsvillkoren (återigen gärna uttryckta i den lokala parametern u)

$$(2) \qquad \begin{cases} \boldsymbol{P}'_i(t_i) = \boldsymbol{P}'_i \\ \boldsymbol{P}'_i(t_{i+1}) = \boldsymbol{P}'_{i+1} \end{cases}$$

som tillsammans med (1) ger fyra villkor ur vilka $\mathbf{a}_j$ kan bestämmas, och man visar lätt att (m a p den lokala parametern u)

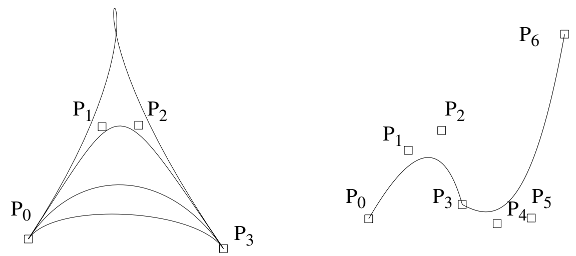
$$(3) \qquad \begin{bmatrix} \boldsymbol{a}_3 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_1 \\ \boldsymbol{a}_0 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_i \\ \boldsymbol{P}_{i+1} \\ \boldsymbol{P}'_i \\ \boldsymbol{P}'_{i+1} \end{bmatrix}$$

I praktiken använder vi denna koefficientmatris för att först bestämma x-komponenterna av $\mathbf{a}_0$ t o m $\mathbf{a}_3$ och därefter y-komponenterna. Detta får upprepas för de olika intervallen.

Den approximerande jämna kurvan plottas sedan genom att man för varje delintervall ritar linjer enligt t ex (om $t_{i+1} = t_i+1$)

$\mathbf{P}_i(t_i)$->$\mathbf{P}_i(t_i+0.1)$->$\mathbf{P}_i(t_i+0.2)$->...->$\mathbf{P}_i(t_i+1.0)$

Nackdelar med metoden: Derivatorna är i praktiken ej kända. Kurvan får kontinuerlig lutning men inte kontinuerlig krökning (andra derivatorna tar skutt).



Figur 3. Till vänster visas några Hermite-kurvor med samma lutning i ändpunkterna. Till höger kubiska Bezier-kurvor med en skarv. De givna punkterna markerade med fyrkanter.

5 Bezierapproximation

I avsikt att få rent lokal påverkan ger vi upp kravet att samtliga punkter skall interpoleras.

För gradtalet 3 bestäms Bezierkurvan genom att man ser på fyra konsekutiva punkter i stöten. Samtliga punkter kallas **styrpunkter** (eng control point). Den första och sista av dessa skall interpoleras och kallas ibland (framför allt i typografisk litteratur) därför även **ankarpunkter**.

Betrakta högra figuren ovan. Tredjegradskurvan mellan $\mathbf{P}_0$ och $\mathbf{P}_3$ kommer enbart att påverkas av dessa två punkter och av de två mellanliggande $\mathbf{P}_1$ och $\mathbf{P}_2$. Novisen skulle råda oss att bestämma tredjegradskurvan som den som interpolerar samtliga fyra punkter (det skulle ju ge de fyra nödvändiga villkoren för bestämning av koefficienterna), men då skulle vi inte säkert få kontinuerlig lutning och inte heller något sätt att uppnå det.

I stället bildar vi med hjälp av styrpunkterna **approximationer** till $\mathbf{P}'_0$ och $\mathbf{P}'_3$ enligt

$\mathbf{P}'_0 = 3(\mathbf{P}_1 - \mathbf{P}_0)$, $\mathbf{P}'_3 = 3(\mathbf{P}_3 - \mathbf{P}_2)$

Varför faktorn 3 och inte 1 som på något sätt verkar naturligare? Jo, om vi låter parametern t löpa mellan 0 och 1 från $\mathbf{P}_0$ till $\mathbf{P}_3$, så är det i brist på något bättre rimligt att låta $\mathbf{P}_1$ motsvara värdet t=1/3 och $\mathbf{P}_2$ motsvara t=2/3. En approximation av derivatan i $\mathbf{P}_0$ ges därmed av $(\mathbf{P}_1 - \mathbf{P}_0)/(1/3)$, som är just högerledet ovan.

Dessa approximationer stoppar man sedan in i (3) i Hermite-avsnittet och erhåller på så sätt ett tredjegradspolynom.

På motsvarande sätt förfares för övriga grupper $\mathbf{P}_{3k}$, ..., $\mathbf{P}_{3k+3}$ om fyra punkter. Motsvarande polynom kommer tydligen att ha linjen mellan $\mathbf{P}_i$ och $\mathbf{P}_{i+1}$ som högertangent i $\mathbf{P}_i$ om i är en multipel av 3. Liknande för vänstertangenten. Man kan därmed lätt interaktivt styra den approximerande kurvans form.

Genom att för varje index i som är en multipel av 3 välja $\mathbf{P}_{i-1}$, $\mathbf{P}_i$ och $\mathbf{P}_{i+1}$ så att de ligger på en linje, kan man ge den approximerande sammansatta kurvan kontinuerlig lutning.

Den polygon som bildas av de fyra styrpunkterna brukar kallas **styrpolygonen**.

Vi kan naturligtvis utifrån den givna beskrivningen räkna fram ett uttryck för det lokala polynomet uttryckt i de givna punkterna. Vi avstår från det nöjet och skriver i stället bara upp det uttryck som i litteraturen vanligen används som definition av Bezierapproximation. Vi antar att parameteravståndet mellan ankarpunkterna är 1. På första kurvsegmentet (mellan $\mathbf{P}_0$ och $\mathbf{P}_3$) gäller

$$\boldsymbol{P}(t) = \sum_{i=0}^{3} B_i(t)\boldsymbol{P}_i$$

där

$$(4) \qquad B_i(t) = \begin{cases} \binom{3}{i} t^i (1-t)^{3-i} & 0 \le t \le 1 \\ 0 & annars \end{cases}$$

Övning: Visa det.

Dessa funktioner, som ju är tredjegradspolynom, finns säkert uppritade i kursböckerna. Allmänt kan vi skriva

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} B_i(t)\boldsymbol{P}_i$$

där för i>3 $B_i(t)$ erhålles ur de fyra första genom en enkel koordinattransformation.

Man kan göra Bezierapproximation med godtyckligt gradtal k (minst 1). Antalet punkter liksom antalet basfunktioner per segment är då k+1. De första basfunktionerna (med index = 0, ..., k) får man genom att i (4) byta de två 3-orna mot gradtalet k. Basfunktionerna, som för övrigt kallas Bernstein polynom (efter en approximationsteoretiker som använde dem långt innan bilmannen Pierre Bezier), är av den givna graden. Bezierapproximation med gradtalet 1 motsvaras av vanlig linjär interpolation. Bezierapproximation med gradtalet 2 används i vissa ritprogram, varvid det mellan två ankarpunkter finns en enda styrpunkt.

Vi ser att basfunktionerna är icke-negativa och att, eftersom de är termerna i binomialutvecklingen av $1 = (t + (1-t))^k$, summan av dem för varje t-värde är 1. Dessa två egenskaper är som vi strax skall se av stor betydelse. Vi kan använda dem för att visa att Bezierapproximation, till skillnad mot global polynominterpolation, inte leder till kraftiga oscillationer.

Uppritning av ett Bezierpolynom kan naturligtvis göras genom att man beräknar polynomets värde i ett antal punkter och drar räta linjer. Det finns ett annat sätt som bara kräver heltalsaritmetik och som används i laserskrivare med PostScript och även i en del ritprogram med utjämningsverktyg. Metoden kallas Casteljaus, men vi går inte in på den här.

6 Konvexa höljet

Om man tittar på ett antal Bezierapproximationer upptäcker man att de enskilda kurvsegmenten alltid ligger inom en månghörning som bildas av fyra eller i vissa fall tre av motsvarande styrpunkter.

Det **konvexa höljet** till en mängd av punkter $\mathbf{Q}_1$, ..., $\mathbf{Q}_M$ är den minsta konvexa mängd som innehåller punkterna. Intuitivt fås det i 2D genom att man spänner ett gummiband runt punkterna. I 3D tar man i stället en gummiduk. Matematiskt utgörs det av alla linjärkombinationer av punkterna som har icke-negativa koefficienter vars summa är 1, dvs av alla punkter **Q** med

$$(5) \qquad \boldsymbol{Q} = \sum_{k=1}^{M} a_k \boldsymbol{Q}_k, \quad \sum_{k=1}^{M} a_k = 1 \qquad och \qquad a_k \ge 0$$

Vid Bezierapproximation gäller enligt ovan för alla punkter på kurvsegmentet mellan $\mathbf{P}_{3k}$ och $\mathbf{P}_{3(k+1)}$ att de är sådana linjärkombinationer. Följaktligen ligger kurvan i det konvexa höljet till $\mathbf{P}_{3k}$, ..., $\mathbf{P}_{3(k+1)}$ (om gradtalet är k i stället för 3, byt de fyra 3-orna i detta stycke mot k).

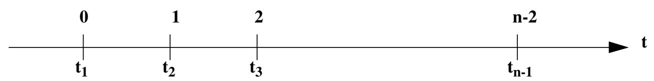
7 Likformiga B-splines

Som tidigare är n+1 punkter $\mathbf{P}_0$, $\mathbf{P}_1$, ..., $\mathbf{P}_n$ givna. Vi vill approximera med en kurva som interpolerar startpunkten $\mathbf{P}_0$ och slutpunkten $\mathbf{P}_n$ och styrs av övriga punkter. Kurvan skall vara styckevis sammansatt av tredjegradskurvor.

För n=3 (4 punkter) kan vi klara det med ett enda kurvsegment $\mathbf{P}_1(t)$, $0 \le t \le 1$. Vi skulle ju t ex kunna använda Bezierapproximation, vilket vår metod i slutänden faktiskt innebär.

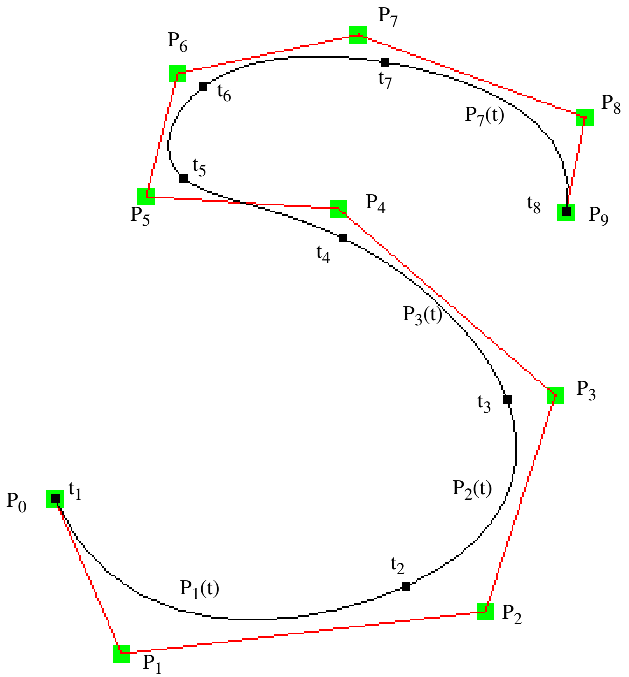
För n=4 (5 punkter) behöver vi två kurvsegment $\mathbf{P}_1(t)$, $0 \le t \le 1$, och $\mathbf{P}_2(t)$, $1 \le t \le 2$. Det första bestäms av $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ och det andra av $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ och $\mathbf{P}_4$.

Allmänt behövs n-2 kurvsegment $\mathbf{P}_i(t)$, $i-1 \le t \le i$, i=1,..., n-2. Vi betecknar med t_i det parametervärde där $\mathbf{P}_{i-1}(t)$ och $\mathbf{P}_i(t)$ går ihop. Speciellt sätter vi $t_1=0$ (första segmentets start) och $t_{n-1}=n-2$ (sista segmentets slut). Man kallar t_i för **skarvvärden** eller **skarvar** (eng knots) och $\mathbf{P}_i(t_i)$ för **skarvpunkter**.



Vi har alltså t_i=i-1, i=1,...,n-1 och antalet skarvar är n-1.

Vi betecknar den totala kurvan med **P**(t), $0=t_1 \le t \le t_{n-1}=n-2$, dvs på intervallet $[t_i,t_i+1]$ är **P**(t)=$\mathbf{P}_i$(t). Jag påminner om att **P**(t) är en vektor, dvs **P**(t)=[x(t), y(t)] i 2D och med ytterligare en komponent i 3D.



Figur 4. Approximation med B-splines med n=10 givna punkter. (GL_SPLINEPAK.c exkl text).

I det allmänna fallet tänker vi oss nu att $\mathbf{P}_i(t)$ bara skall påverkas av de fyra punkterna $\mathbf{P}_{i-1}$, $\mathbf{P}_i$, $\mathbf{P}_{i+1}$ och $\mathbf{P}_{i+2}$. Inflytandet skall naturligtvis variera med t. Stärkta av framgångarna med Bezierapproximationen gör vi därför en ansats av formen

$$(6) \qquad \mathbf{P}_i(t) = B_{i-1}(t)\mathbf{P}_{i-1} + B_i(t)\mathbf{P}_i + B_{i+1}(t)\mathbf{P}_{i+1} + B_{i+2}(t)\mathbf{P}_{i+2}$$

Det återstår att bestämma **blandnings/bas-funktionerna** $B_i(t)$ för i=0, 1, 2, ..., n. Eftersom vi vill ha lokal påverkan, måste vi se till att $B_i(t)$ bara är skild från noll nära $t=t_i$. Därför kan $B_i(t)$ omöjligen vara ett tredjegradspolynom. Men vi kan låta den vara hopskarvad av tredjegradspolynom. Vi försöker åstadkomma detta genom att välja $B_i(t)$ som ett tredjegradspolynom på vart och ett av intervallen och sådant att $B_i(t)$ är 0 utanför intervallet $[t_{i-2},t_{i+2}]$. Det visar sig vara möjligt att göra detta så att $B_i(t)$ globalt har kontinuerliga derivator upp till ordning 2. Den uppmärksamme läsaren ser att bl a $\mathbf{P}_1(t)$ skapar ett problem. Då behövs $B_0(t)$, som skall vara 0 utanför intervallet $[t_{-2},t_2]$. Men några skarvar t_{-2}, t_{-1}, och t_{-0} har vi ju inte. Vi återkommer till detta om en liten stund.

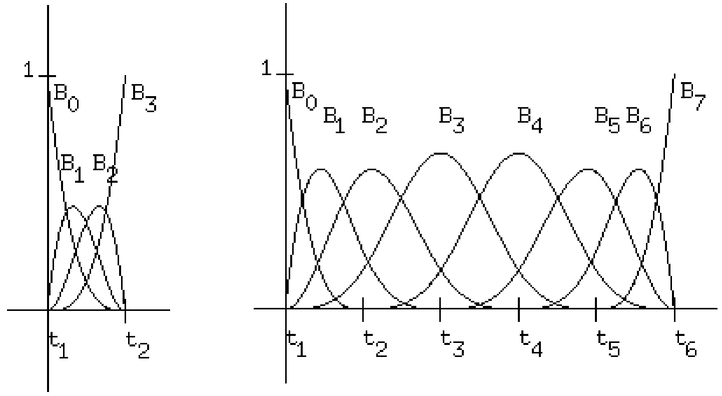
Vi uppnår lokal påverkan och samtidigt kontinuerlig krökning. Det pris vi får betala är samtliga punkter inte interpoleras. Det betyder dock inte så mycket i ett modelleringssystem (som t ex CAD-program), där man främst eftersträvar interaktivt arbete med geometriskt gripbara parametrar.

Testfråga: Varför modellerar man inte sådan här styckvisa polynom med de lokala polynomens koefficienter?

Den resulterande kurvan, som kan skrivas

$$(7) \qquad \mathbf{P}(t) = \sum_{i=0}^{n} B_i(t)\mathbf{P}_i$$

där högst fyra termer ger något bidrag för givet t, får då kontinuerlig lutning och krökning. Figuren nedan visar basfunktionerna.



Figur 5. Basfunktionerna. Till vänster för fallet 4 punkter (4 basfunktioner) och till höger för 8 punkter (8 basfunktioner).

Det är basfunktionerna som heter **B-splines**. Benämningen betyder Base Splines med innebörden att de utgör en bas för all styckevis approximation med polynom. Vi ser av figurerna att deras utseende varierar något beroende av antalet punkter. Detta visar sig dock bara gälla de yttre. De inre som t ex B_3 och B_4 i högra figuren har en enhetlig form och kan beskrivas med $B_i(t) = B(t-t_i)$ (om man som vi antar att skillnaden mellan successiva skarvar är 1). Våra krav uppfylles om B(t) är noll utanför [-2,2] och har den regularitet som vi önskar oss av $B_i(t)$.

Av figuren framgår att basfunktionerna är icke-negativa. Vi kommer även att se till att deras summa för varje t är 1. Detta betyder att kurvsegmentet $\mathbf{P}_i(t)$ precis som när det gällde Bezierapproximation ligger i det konvexa höljet till de fyra punkterna $\mathbf{P}_{i-1}$, $\mathbf{P}_i$, $\mathbf{P}_{i+1}$ och $\mathbf{P}_{i+2}$. Speciellt kommer approximationen att bli en rät linje genom punkterna om dessa från början ligger på en sådan.

Låt oss nu räkna på de kubiska basfunktionerna. Den normerade basfunktionen B(t) skall vara 0 utanför intervallet |t|<2 och skall vara kontinuerlig med kontinuerliga derivator upp till ordning 2.

B(t) tillåts vara sammansatt av fyra polynom med gradtalet 3. Vi har därför 16 storheter - koefficienterna - för vars bestämning det behövs 16 villkor. Problemet att bestämma B(t) kan således ses som ett ekvationssystem med 16 obekanta. Med en aning klurighet kan vi dock komma billigare undan. Först och främst reducerar symmetrin kring t=0 det hela till 8 storheter. Vi har villkoren

B(2) = 0, B'(2) = 0, B"(2) = 0 p g a regularitetskraven
B'(0) = 0 p g a symmetrin
$B_-(1) = B_+(1)$, $B'_-(1) = B'_+(1)$, $B''_-(1) = B''_+(1)$ p g a regularitetskraven
B(0) + 2B(1) = 1 eftersom summan av basfunktionerna skall vara 1

Detta ger 8 villkor för de 8 parametrar som hör ihop med de väsentligen två tredjegradspolynom som skall bestämmas. Med ett litet trick kan vi komma undan enklare. De olika kraven gör att på intervallet [1,2] måste B(t) ha formen $C(2-t)^3$. Vi har därmed bara 5 parametrar kvar att bestämma. På motsvarande sätt är det lämpligt att ansätta ett tredjegradspolynom i 1-t på intervallet [0,1]. Man visar tämligen lätt och rutinmässigt att

$$B(t) = \begin{cases} \frac{1}{6}(2-|t|)^3 & 1 \le |t| \le 2 \\ \frac{1}{6}[1 + 3(1-|t|) + 3(1-|t|)^2 - 3(1-|t|)^3] & |t| \le 1 \\ 0 & 2 \le |t| \end{cases} \tag{8}$$

Vi kan nu med hjälp av formel (8) analytiskt övertyga oss om att skarvpunkten $\mathbf{P}_i(t_i)$ i allmänhet inte sammanfaller med den givna punkten P_i. (8) ger nämligen

$$\boldsymbol{P}_i(t_i) = B(-1)\boldsymbol{P}_{i-1} + B(0)\boldsymbol{P}_i + B(1)\boldsymbol{P}_{i+1} + B(2)\boldsymbol{P}_{i+2} = \frac{1}{6}\boldsymbol{P}_{i-1} + \frac{4}{6}\boldsymbol{P}_i + \frac{1}{6}\boldsymbol{P}_{i+1}$$

som bara undantagsvis sammanfaller med $\mathbf{P}_i$. Vi ser emellertid härav att $\mathbf{P}_i(t_i)$ ligger i det konvexa höljet till de tre punkterna $\mathbf{P}_{i-1}$, $\mathbf{P}_i$ och $\mathbf{P}_{i+1}$.

Om vi utgår från figurerna ovan är det på motsvarande lätt att beräkna de avvikande basfunktionerna, t ex B_0, B_1 och B_2 i högra delen av figuren. Eftersom vi vill ha interpolation i startpunkten $\mathbf{P}_0$, dvs $\mathbf{P}_1(0) = \mathbf{P}_0$, måste med den gjorda ansatsen $B_0(0)=1$ och $B_1(0)=B_2(0)=0$. Om som i figuren $B_0(t)$ bara är skild från 0 på [0,1] och har två kontinuerliga derivator måste därför $B_0(t)=(1-t)^3$. De båda andra bjuder en aning mera motstånd.

Det finns ett tilltalande och enhetligt sätt att beskriva samtliga basfunktioner. En inre basfunktion $B_i(t)$ kan sägas vara bestämd av **skarvföljden** (**skarvvektorn**, eng. knot vector) $[t_{i-2},t_{i-1},t_i,t_{i+1},t_{i+2}]$ eller med våra förutsättningar [i-3,i-2,i-1,i,i+1]. Den är "centrerad" kring t_i som i vårt fall är i-1. De inre basfunktionerna har bestämts så att de är 0 utanför intervallet $[t_{i-2},t_{i+2}]$. Vi kan använda samma formulering om övriga genom att införa 3 extra skarvar $t_{-2} = t_{-1} = t_{-0} = 0$ före de andra och 3 extra $t_n = t_{n+1} = t_{n+2} = n-2$ i slutet. $B_0(t)$ bestäms då av [0,0,$\underline{0}$,0,1], medan $B_1(t)$ bestäms av [0,0,$\underline{0}$,1,2] (om antalet punkter är minst 5, annars [0,0,$\underline{0}$,1,1]) och $B_2(t)$ bestäms av [0,0,$\underline{0}$,1,2] (om antalet punkter minst 5, annars [0,0,$\underline{1}$,1,1]). På motsvarande sätt i den andra änden. Den ursprungliga skarvföljden om n-1 värden $[t_1,t_2,...,t_{n-1}]$ utökas på detta vis med 6 skarvar, dvs omfattar totalt (n+1)+4 skarvar eller 4 mer än antalet punkter. För interpolation i ändpunkterna skall de fyra första och de fyra sista i den nya följden vara sinsemellan lika. Vi har nu förklaringen till förfarandet i OpenGL.

Exempel: För sex punkter har vi den ursprungliga skarvföljden [0,1,2,3] och den utökade [0,0,0,0,1,2,3,3,3,3]. Basfunktionen $B_0(t)$ ges av de fem första värdena, nästa av de därpå följande fem, o s v.

Det finns en rekursiv formel för beräkning av godtycklig basfunktion $B_i(t)$ givet en skarvföljd med fem punkter, varav vissa eventuellt lika. Denna formel är utgångspunkten för resonemangen om B-splines i såväl HB (sid 335) som FvD. Vi tar upp den längre fram.

Fortfarande har vi inte beräknat $B_1(t)$ och $B_2(t)$. Men vi lägger inte ned manuell möda på det, eftersom det låter sig göras smärtfritt med t ex matematikprogrammet Maple (startas med kommandot *xmaple*).

Det går till så här. Först får man läsa in ett bibliotek med

```
> readlib(bspline);
```

Man anropar sedan en funktion bspline med tre parametrar. Den första anger gradtalet. Den andra önskat parameternamn och den sista skarvföljden. T ex får vi med

```
> b0:=bspline(3,t,[0,0,0,0,1]);
```

funktionen $B_0(t)$ (efter kopiering från Maple till FrameMaker och litet redigering)

```
b0 := PIECEWISE([0, t < 0],[-t^3+3*t^2-3*t+1, t < 1],[0, 1 <= t])
```

Genom att skriva

```
> factor(b0);
```

får vi en snyggare form

```
PIECEWISE([0, t < 0],[-(t-1)^3, t < 1],[0, 1 <= t])
```

På liknande sätt kan vi beräkna de önskade $B_1(t)$ och $B_2(t)$:

```
> b1:=factor(bspline(3,t,[0,0,0,1,2]));
b1 := PIECEWISE([0, t < 0],[1/4*t*(7*t^2-18*t+12), t < 1],
                [-1/4*(t-2)^3, t < 2],[0, 2 <= t])
> b2:=factor(bspline(3,t,[0,0,1,2,3]));
b2 := PIECEWISE([0, t < 0],[-1/12*t^2*(11*t-18), t < 1],
        [-3/2+9/2*t-3*t^2+7/12*t^3, t < 2],[-1/6*(t-3)^3, t < 3],[0, 3 <= t])
```

Om vi även beräknar $B_3(t)$ som b3, kan vi - eftersom Maple arbetar symboliskt - förvissa oss om att summan $B_0(t)+B_1(t)+B_2(t)+B_3(t)$ blir 1 på intervallet [0,1] (om vi nu litar på Maple):

```
> simplify(b0+b1+b2+b3);
PIECEWISE([0, t < 0],[1, t <= 1],....
[7/6-1/2*t-1/6*t^3+1/2*t^2, t <= 2],[-17/6+11/2*t+1/3*t^3-5/2*t^2, t <= 3],
[-8*t+32/3-1/6*t^3+2*t^2, t <= 4],[0, 4 < t])
```

Man kan konvertera uttrycken så att de lättare kan plockas in i vanlig programkod. T ex

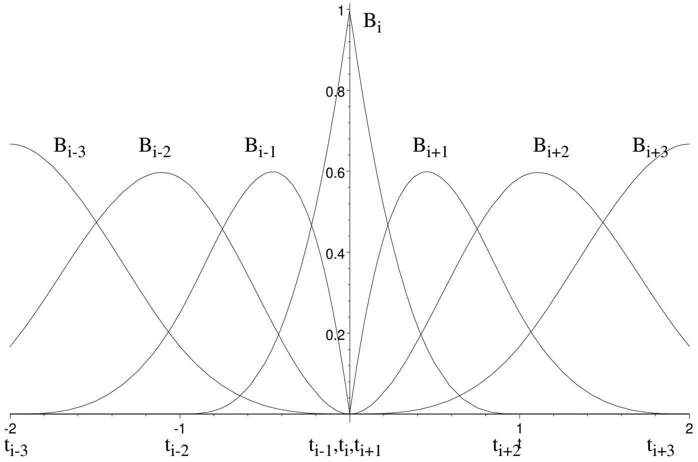
```
> s:=convert(b1,procedure);
s := proc () piecewise(t < 0,0,t < 1,1/4*t*(7*t^2-18*t+12),
        t < 2,-1/4*(t-2)^3,2 <= t,0)
     end
```

Naturligtvis går det att rita upp basfunktionerna om vi vill i Maple. Men nu får det vara nog.

8 Olikformiga B-splines

Inget hindrar att skarvvärdena t_i inte är ekvidistanta. Det enda problemet är att det blir värre att bestämma de olika basfunktionerna. Med olikformiga B-splines avses situationer då skarvpunkterna inte är ekvidistanta. Vi uppehåller oss inte vid det allmänna fallet utan går in på ett fall som är av stort praktiskt intresse och som vi redan mött nämligen att ett antal skarvar är lika. Vi använde detta i ändpunkterna för att få interpolation. Samma sak kan göras i inre punkter och bl a leder det till att Bezierapproximation kan uppfattas som ett specialfall av B-splinesapproximation.

Vad händer om vi låter några skarvpunkter sammanfalla, men bibehåller kravet att en basfunktion $B_i(t)$ skall vara noll utanför $[t_{i-2},t_{i+2}]$? Vi får då färre samband och kan inte uppfylla de villkor som vi önskar, dvs vi måste reducera regularitetskraven. Avstånden mellan skarvarna blir nu 0 eller 1. En sådan situation visas i följande figur med tre skarvvärden som sammanfaller för $t = t_i$. Basfunktionen $B_i(t)$ sträcker sig nu över bara två riktiga delintervall. Eftersom $B_{i-1}(t)$ skall vara 0 för $t \geq t_{i+1} = t_i$, måste $B_{i-1}(t_i) = 0$. Samma gäller för B_{i+1} och B_{i+2}. Följaktligen måste (summakravet) $B_i(t_i) = 1$.



Figur 6. Trippelskarv. Beräknad och ritad med Maple.

Vi vill ta fram ett analytiskt uttryck för basfunktionen $B_i(t)$ för fallet i figuren. Låt oss för enkelhets skull anta att avstånden mellan konsekutiva icke-överlappande skarvar är 1. Vi kan då normalisera och anta att de fem intressanta skarvpunkterna är [-1,0,0,0,1]. $B_i(t)$ skall vara ett tredjegradspolynom på [-1,0] och ett annat på [0,1]. Rimligen har vi symmetri, varför det räcker att se på [0,1]. Vi har då att (vi slopar indexet)

$B(t) = a(1-t) + b(1-t)^2 + c(1-t)^3$

Kravet kontinuerlig andraderivata i 1 ger att a = b = 0. Och B(0) = 1 ger att

$B(t) = (1-t)^3$

I t_i (=0) får vi nu kontinuitet enbart hos funktionen, inte hos derivatorna.

Allmänt gäller att varje ökning av en skarvs multiplicitet sänker regulariten hos basfunktionerna som berör punkten ett steg. Och det just i den punkten. Motsvarande sänkning drabbar naturligtvis - utom i undantagsfall - vår approximerande kurva.

Det är inte svårt att visa att trippelskarvar gör att man får interpolation i motsvarande styrpunkt. Det är inte heller svårt att visa att intilliggande trippelskarvar gör att motsvarande kurvsegment är en Bezier-approximation. Men vi avstår från dessa överläggningar och konstaterar bara att Bezier-approximation därmed kan ses som ett specialfall av B-splinesapproximation.

När multipliciteten är 2 har vi kontinuerlig derivata. När den är 3 har vi kontinuerlig funktion (och även kontinuerlig derivata vid lämplig placering av punkterna). Hur går det om den är 4? Jo, då blir den resulterande kurvan inte ens kontinuerlig. Således kan vi med en B-splineapproximation hantera även sönderbrutna kurvor. För närmare utredning av detta och belysande figur se FvD.

9 Andra gradtal

Man kan använda godtyckligt gradtal k vid B-splinesapproximation. Basfunktionerna skall vara sådana att de har kontinuerliga derivator till och med ordningen k-1. För udda gradtal k=2m-1 gäller att $B_i(t)$ är skild från noll bara på intervallet $[t_i-m, t_i+m]$ och för jämna gradtal k=2m på intervallet $[t_i-m-1, t_i+m]$.

I fallet k = 1 skall vi ha en styckevis linjär funktion som är noll om $1 \leq |t|$. Vi får villkoren för den normerade basfunktionen

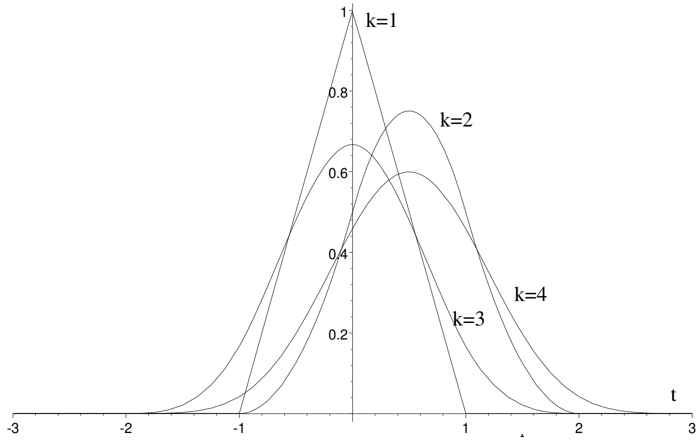
B(1) = 0, B(0) = 1 (enligt summaformeln ovan)

dvs med utnyttjande av symmetrin

$B(t) = 1 - |t|$ för $|t| \leq 1$ och 0 annars.

I följande figur har vi ritat upp några normerade basfunktioner B(t) för olika gradtal med hjälp av Maple:

```
s1:=bspline(1,t,[-1,0,1]); s2:=bspline(2,t,[-1,0,1,2]);
s3:=bspline(3,t,[-2,-1,0,1,2]); s4:=bspline(4,t,[-2,-1,0,1,2,3]);
plot({s1,s2,s3,s4},t=-3..3, color=black);
```



Figur 7. Några B(t) kurvor för gradtalen k=1,2, 3 och 4.

10 NURBS

Alla de approximationsmetoder vi har tittat på hittills ger affint invarianta kurvor. Med detta menas att det spelar ingen roll om man först transformerar de givna punkterna och sedan beräknar approximationen eller transformerar kurvan i sin helhet. Det senare vore en beräkningsmässigt förödande ineffektiv metod. En affin transformation är en icke-singulär linjär transformation. Låt oss här beteckna den med M. Låt P(t) vara en punkt på vår kurva. Då är

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} C_i(t) \boldsymbol{P}_i \qquad (9)$$

där $C_i(t)$ är vissa koefficienter. Detta gäller även vid kubisk splineinterpolation, men då är de flesta skilda koefficienterna skilda från 0. Eftersom transformationen M är linjär får vi

$$(10) \qquad M\boldsymbol{P}(t) = \sum_{i=0}^{n} C_i(t) M\boldsymbol{P}_i$$

som visar det påstådda. Vi kan sålunda skala, rotera och translatera våra kurvor genom att göra motsvarande operationer på de givna punkterna och sedan göra t ex en ny Bezierapproximation.

Ingen av metoderna ger däremot kurvor som är invarianta under en perspektivtransformation, dvs klarar övergången från 3D till 2D. Vi ger inte något exempel utan nöjer oss med påståendet. Så kallade NURBS (Non Uniform Rational B-Splines) som är en obetydlig generalisering av vanliga icke-likformiga B-splines ger däremot det. De har också förmågan att representera vanliga kägelsnittskurvor (ellipser, parabler och hyperbler) exakt, vilket inte heller de tidigare metoderna klarar. Vidare ger NURBS ytterligare modelleringsparametrar. Vi skall här inte bevisa eller ens motivera de två första egenskaperna utan anger bara formen och ser på exempel.

B-splinesapproximation (kallas i t ex FvD non-rational, vilket känns tungt) har hittills inneburit

$$(6) \qquad \boldsymbol{P}_i(t) = B_{i-1}(t)\boldsymbol{P}_{i-1} + B_i(t)\boldsymbol{P}_i + B_{i+1}(t)\boldsymbol{P}_{i+1} + B_{i+2}(t)\boldsymbol{P}_{i+2}$$

Motsvarande NURBS-approximation är

$$(11) \qquad \boldsymbol{P}_i(t) = \frac{\sum_{j=i-1}^{i+2} w_j B_j(t) \boldsymbol{P}_j}{\sum_{j=i-1}^{i+2} w_j B_j(t)}$$

där talen w_j är viktvärden som normalt är icke-negativa (annars kan det bl a bli problem i nämnaren). Om alla $w_j = 1$ har vi det vanliga fallet. Beräkningsmässigt innebär NURBS inga komplikationer. Vi ersätter bara i kalkylerna $B_j(t)$ med

$$\frac{w_j B_j(t)}{\sum_s w_s B_s(t)}$$

Effekten av att ha w_j annorlunda än 1 är att $P_i(t_i)$ dras närmre $\boldsymbol{P}_i$ när w_i är större än 1 och växer.

11 Beräkningsaspekter för B-splinesapproximation

På intervallet $[t_i, t_{i+1}]$ ges B-splinesapproximation $\boldsymbol{P}_i(t)$ av formel (6). Uppritningen av det kurvsegmentet görs genom att vi beräknar och sammanbinder enligt t ex

$\boldsymbol{P}_i(t_i) \to \boldsymbol{P}_i(t_i+0.1) \to \boldsymbol{P}_i(t_i+0.2) \to \ldots \to \boldsymbol{P}_i(t_i+1)$

Beräkning av $\boldsymbol{P}_i(t)$ för ett visst t-värde t kräver beräkning av $B_{i-1}(t)$, $B_i(t)$, $B_{i+1}(t)$ och $B_{i+2}(t)$.

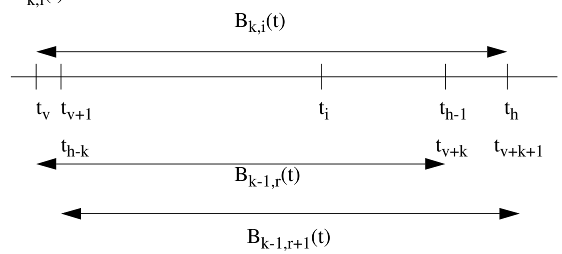
Låt oss först se på likformiga fallet och utan att ta hänsyn till ändpunktsproblemen. Eftersom alla basfunktionerna då kan uttryckas i en gemensam, $B_i(t) = B(t-t_i)$, där $B(t) = 0$ för $|t|>2$, räcker det att beräkna B(t) i ett litet antal punkter som lagras i en vektor. Beräkningen av $P_i(t)$ för de förutsedda t-värdena blir därmed nästan ögonblicklig.

Om vi har multipla skarvar (som ju klarar ändpunkterna) men för övrigt likformighet, betyder det bara att man måste införa ytterligare några vektorer med värden för modifierade basfunktioner. Samma gäller om man har ett litet antal olika intervall-längder.

Vi nämnde att man kan införa basfunktioner även för andra gradtal k än 3. Låt oss använda beteckningen $B_{k,i}(t)$ för dessa. Basfunktionerna skall vara uppbyggda av k-tegradspolynom och vara sådana att de har kontinuerliga derivator till och med ordningen k-1 (i frånvaro av multiplicitet, annars lägre). De är skilda från noll på ett intervall med k+2 skarvpunkter. För udda gradtal $k=2m-1$ gäller att $B_i(t)$ är skild från noll bara på intervallet $[t_i-m, t_i+m]$ och för jämna gradtal $k=2m$ på intervallet $[t_i-m-1, t_i+m]$ (i figur 7 olyckligtvis litet annorlunda).

Anm. Vissa författare använder andra numreringar av basfunktionerna, t ex att $B_{k,i}(t)$ hör till $[t_i, t_{i+k+1}]$. Åter andra avstår från all numrering och skriver $B_{k,[\text{uppräkning av skarvpunkterna}]}(t)$. Det kan också nämnas att somliga låter k stå för gradtalet+1.

Det finns ett inte ointressant rekursivt samband mellan basfunktionerna av olika ordningar k. Vi skall inte bevisa det utan bara notera det med ett par kommentarer. Formeln uttrycker $B_{k,i}(t)$ i $B_{k-1,i}(t)$ och $B_{k-1,i+1}(t)$ om k är udda (annars byt de två sista i mot i-1). I följande figur är $[t_v, t_h]$ det intervall på vilket $B_{k,i}(t)$ är skild från noll.



Figur 8. Figur som belyser rekursionsformeln. I figuren är r=i om k udda och r=i-1 om k jämnt. Vidare är h=i+m och v=i-m (om k udda) resp v=i-m-1 (om k jämnt), där k=2m (om k jämnt) och k=2m-1 (om k udda).

Med beteckningarna i figuren gäller

$$(12) \qquad B_{k,i}(t) = \frac{t - t_v}{t_{v+k} - t_v} B_{k-1,r}(t) + \frac{t_h - t}{t_h - t_{h-k}} B_{k-1,r+1}(t)$$

Det är klart att gradtalet höjs med 1 och att stödet (det intervall där en funktion är skild från noll) utvidgats. På grund av faktorn $(t-t_v)$ ökas regulariteten också med 1 i vänstra ändpunkten. Det återstår egentligen bara att visa att motsvarande gäller i övriga skarvar men det avstår vi ifrån.

Givet $B_{0,i}(t) = 1$ för $t_{i-1} \le t < t_i$ och 0 för övrigt, kan man därför beräkna basfunktionsvärden för godtyckligt gradtal och för likformiga likaväl som olikformiga skarvföljder. Snabbt går det däremot inte. Maple använder sig av denna formel, när det tar fram formler för B-splines. Kravet

$$\sum_i B_{0,i}(t) = 1$$

gör att $B_{0,i}(t)$ bara kan vara 1 i endera t_{i-1} (vilket vi valt) eller t_i.

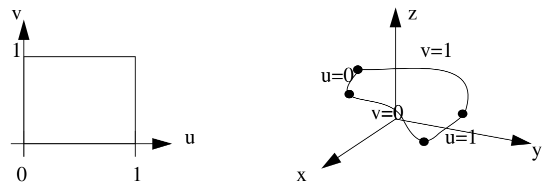
Det finns en komplikation. Säg att vi vill beräkna $B_{3,i}(t)$ när t_i är en trippelskarv, t ex för skarvföljden $[-1,0,0=t_i,0,1]$. Då behövs B_2 för bl a $[-1,0,0,0]$, som i sin tur behöver B_1 för $[-1,0,0]$ och för $[0,0,0]$. För den senare är emellertid nämnaren 0! Den enkla utvägen är att strunta i de termer där nämnaren är 0.

12 Ytor

12.1 Allmänt

I avsnitt 1 beskrev vi en tolkning av parameterframställningen $\mathbf{P}=\mathbf{P}(t)$, $0 \le t \le 1$ för ett kurvstycke.

På motsvarande sätt kan parameterframställningen $\mathbf{P}=\mathbf{P}(u,v)$, $0 \le u,v \le 1$, för ett ytsegment (eng. patch) ses som avbildningen av en kvadrat. Se följande figur.



Figur 9. Ytsegment

Exempel: Plan yta.
Givet en punkt $\mathbf{P}_{00}$ och två icke-parallella vektorer $\mathbf{S}$ och $\mathbf{T}$ spänner
$\mathbf{P} = \mathbf{P}(u,v) = \mathbf{P}_{00} + u\mathbf{S} + v\mathbf{T}$, $0 \le u,v \le 1$,
ett plant ytsegment med fyra hörn. Vi kan alternativt starta med tre givna punkter $\mathbf{P}_{00}$, $\mathbf{P}_{10}$, $\mathbf{P}_{01}$, varefter $\mathbf{S}= \mathbf{P}_{10}-\mathbf{P}_{00}$ och $\mathbf{T}=\mathbf{P}_{01}-\mathbf{P}_{00}$.

12.2 Segmentvis approximation

Vi ger oss nu in på motsvarigheten till styckevis approximation för kurvor. Vi tänker oss att den lokala approximationen skall göras på ett segment motsvarande $0 \le u,v \le 1$ (precis som vi i kurvfallet såg på ett segment $0 \le t \le 1$). En riktig känsla för arbete med ytor får man inte gratis.

Motsvarigheten till linjär interpolation skulle då vara att vi bestämde ett plan som gick genom de fyra punkterna $\mathbf{P}(u,v)$, $u=0,1, v=0,1$. Men vi har då ett villkor för mycket (jfr exemplet i slutet av 14.1). Vill vi approximera lokalt med plan måste vi stället ha trianglar som bassegment.

I stället blir den naturliga approximationen en bilinjär (betyder att den är linjär i varje parameter för sig) yta

$$\boldsymbol{P}(u,v) = \sum_{0 \le i,\, j \le 1} a_{i,\,j} u^i v^j,\ 0 \le u,\, v \le 1$$

Vi har här fyra fria parametrar, vilka går att bestämma ur de fyra geometriska villkoren.

Men precis som när det gäller kurvor vill man ha bättre (läs mjukare) approximationer. Nästa steg är bikubiska ytor

$$\boldsymbol{P}(u,v) = \sum_{0 \le i,\, j \le 3} a_{i,\,j} u^i v^j,\ 0 \le u,\, v \le 1$$

Vi har nu hela 16 fria parametrar, vilka går att bestämma om vi förutsätter att $\mathbf{P}$, $\mathbf{P}_u$, $\mathbf{P}_v$ och $\mathbf{P}_{uv}$ är givna i de fyra hörnen. Denna typ av approximation kallas Fergusonapproximation och är alltså motsvarigheten till Hermiteapproximation. Men den är tydligen än värre att arbeta med.

Vi övergår därför till våra favoriter Bezier och B-splines från kurvfallet.

En variant är att konstruera en bi-Bezieryta (vanligen utelämnas bi-), dvs (vi är bara intresserade av det kubiska fallet)

$$P(u,v) = \sum_{0 \le i,\, j \le 3} B_i(u) B_j(v) \boldsymbol{P}_{i,\,j},\ 0 \le u,\, v \le 1$$

där $\mathbf{P}_{ij}$ är styr- och ankarpunkterna. Man har fyra sådana punkter i varje koordinatriktning, dvs totalt 16 st. Ytan går genom enbart fyra av dessa nämligen $\mathbf{P}_{00}$, $\mathbf{P}_{03}$, $\mathbf{P}_{30}$ och $\mathbf{P}_{33}$. Se figurer i böckerna.

En annan är att konstruera en bi-B-splineyta (vanligen utelämnas bi-), dvs (vi är bara intresserade av det kubiska fallet)

$$\boldsymbol{P}(u,v) = \sum_{0 \le i,\, j \le n} B_i(u) B_j(v) \boldsymbol{P}_{i,\,j},\ 0 \le u,\, v \le n-2$$

där $\mathbf{P}_{ij}$, $0 \le i,j \le n$, är styrpunkterna. Vi har här antagit att antalet punkter är lika i de båda "riktningarna", men de kan naturligtvis vara olika. Se återigen figurer i böckerna.

13 Litet som tiden inte räckte till

Vi har nämnt att valet av parametrisering har viss betydelse. Det skulle man kunna säga mera om.

Vi har också sagt något i stil med att "kontinuerlig krökning" är ekvivalent med kontinuerlig andraderivata. Det är nu inte riktigt sant.

Man kan ha kontinuerlig lutning och krökning utan att ha kontinuerlig derivata resp andra derivata. För detta finns ett begrepp som kallas **G-kontinuitet** (s k geometrisk kontinuitet). Detta kan användas för att få ytterligare en parameter som påverkar kurvformen. Den intresserade hänvisas till FvD. HB sid 318-319 är däremot alldeles för kortfattad.

Vidare garanterar inte kontinuerlig derivata (eller kontinuerlig andraderivata) att man har kontinuerlig lutning (resp kontinuerlig krökning). Problem med lutningen kan uppstå i punkter där både x'(t) och y'(t) är 0. Även nu hänvisas den intresserade till FvD.

Till slut ett par referenser.
Farin: Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Academic Press 1990. Trots sin titel en halv-teoretisk bok, som jag själv haft visst nöje av. Delvis lättläst, men innehåller också svårgenomträngligt material. Innehåller 11 sidor där Bezier beskriver bakgrunden till sina kurvor.
IEEE-tidskriften Computer Graphics and Applications innehåller ofta läsvärt och välpresenterat material. Ett exempel.
Les Piegl: On NURBS: A Survey, January 1991