

Splines

Erik Leppen

14 juli 2006

Inhoudsopgave

1	Inleiding	2
2	Lagrange-interpolatie	3
2.1	Eendimensionale interpolatie	3
2.2	Lagrange-interpolatie	5
2.3	Oscillaties	6
3	Meerdimensionale interpolatie	7
3.1	Parametrisaties	7
3.2	Approximatie	8
4	Bézierkrommen	10
4.1	Twee punten	10
4.2	Drie punten	10
4.3	Het algemene geval	11
4.4	De afleiding van de formule	12
4.5	Afgeleiden van Bézierkrommen in de eindpunten	15
5	Samengestelde Bézierkrommen	17
6	B-splines	18
6.1	B-Splines van graad 3	18
6.2	De formule voor de B-spline	20
6.3	Differentieerbaarheidseigenschappen van B-splines	21
7	Splines in de praktijk	23
7.1	Eigenschappen van B-splines	23
7.2	Toepassingen	24
8	Referenties	25

1 Inleiding

Tegenwoordig worden bij steeds meer taken computers als hulpmiddel gebruikt. Sommige van deze taken zijn maatschappelijk zeer invloedrijk, denk aan het ontwerpen van bruggen of het maken van weersvoorspellingen. Sommige doeleinden zijn maatschappelijk iets minder relevant, denk dan onder meer aan het ontwerpen van computergames. Ongeacht de maatschappelijke belangen is het soms nodig om bepaalde krommen in de computer te kunnen maken en bekijken. Helaas voor ons werkt een computer altijd met getallen van beperkte precisie en kunnen we dus nooit een *perfecte* kromme genereren. We kunnen al niet direct een perfecte cirkel maken op een computer: als we deze als kromme zouden willen opslaan, moeten we deze laten bestaan uit een zeer groot aantal kleine lijnstukjes, die aan de uiteinden verbonden zijn. Echter, er is een slimmere manier om een cirkel $c \in \mathbb{R}^2$ in een computer op te slaan, namelijk door het middelpunt en de straal te onthouden, of het middelpunt en een punt op de cirkel op te slaan. Zo kunnen we met slechts twee "gegevens" een cirkel onthouden, dat is dus ófwel een punt en een getal, ófwel twee punten. Een cirkel in driedimensionale ruimte zouden we kunnen opslaan met behulp van drie punten: het middelpunt, een punt op de cirkel en een niet collineair punt in het vlak waarin de cirkel ligt.

Dit geeft ons te denken of we niet meer krommen op een handige manier kunnen opslaan, in plaats van ze te laten bestaan uit vele kleine lijnstukjes. Bijvoorbeeld ellipsen, parabolen, hyperbolen, of bepaalde oppervlakken. Al deze krommen "liggen vast" als een aantal punten gegeven zijn. Voor een parabool bijvoorbeeld, definiëren we het brandpunt, en met twee andere punten definiëren we de richtlijn.

Echter, we willen ons niet beperken tot een kleine collectie krommen (cirkels, kegelsneden, enzovoorts). De vraag is dus, gegeven een *willekeurige* kromme, kunnen we deze efficiënt opslaan? Het antwoord: we kunnen de gegeven kromme niet efficiënt opslaan, tenzij we, in plaats van de kromme zelf, tevreden zijn met een benadering. Wat we willen, is een aantal typen krommen invoeren die geschikt zijn om willekeurige gegeven krommen te benaderen. Dit willen we zodanig doen, dat deze benaderingskrommen te definiëren zijn door slechts een beperkt aantal punten, en zodanig dat de wiskundige eigenschappen prettig zijn. Wat dit inhoudt, komt later aan de orde.

Een ander voordeel van een dergelijke manier van opslaan is dat de precisie waarmee de kromme getekend kan worden, niet vast ligt. Denk weer aan de cirkel. Als we gewoon duizend punten op de cirkel zouden opslaan en die met lijnstukjes zouden verbinden, dan is het later niet meer mogelijk om daar tienduizend punten van te maken, als bijvoorbeeld blijkt dat duizend te weinig was. De computer weet namelijk niet dat het een cirkel moet voorstellen en kan dus geen

extra tussenpunten plaatsen. Als we echter alleen middelpunt en straal opslaan, kunnen we bij het tekenen kiezen welke precisie we willen: we kunnen duizend punten uitrekenen en die met lijntjes verbinden, maar evengoed tienduizend of een miljoen. In de games wordt dit ook gedaan: er zijn verschillende grafische instellingen mogelijk.

Het blijkt dat het inderdaad mogelijk is om “willekeurige” krommen op te slaan in een beperkt aantal punten. De krommen die daarvoor worden gebruikt, heten **splines**. Uiteraard kunnen we niet àlle krommen op deze manier maken, maar we kunnen krommen wel willekeurig dicht benaderen. Cirkels en kegelsneden zijn exact te maken, echter de splines die daarvoor nodig zijn, zijn ingewikkelder dan de splines die we hier zullen behandelen.

Ook moeten we even stilstaan met wat we bedoelen met “opslaan”. Een kromme *opslaan met punten* betekent hier dat zonder andere informatie dan deze punten en hun volgorde, we de kromme kunnen verkrijgen. Uiteraard is het hierbij toegestaan dat twee sets punten in dezelfde kromme resulteren.

We kijken eerst naar interpolatie, een techniek die gebruikt wordt om tussenwaarden van functies te schatten.

2 Lagrange-interpolatie

2.1 Eendimensionale interpolatie

Interpolatie is het schatten van tussenpunten van functies die we niet overal exact kennen. Bijvoorbeeld, we kennen de functiewaarden in de punten $x = 1, 2, 3$ en 6 , maar zijn geïnteresseerd in de functiewaarde in $x = 4$. We doen dat door een interpolatie toe te passen op deze vier waarden door een polynoom dat precies door deze punten gaat. Zo krijgen we een kromme die volgt uit deze vier punten, die we het **interpolatiepolynoom** van de functie noemen. We kijken naar een voorbeeld.

Voorbeeld 2.1.1 *De punten $(1, 3)$, $(2, 2)$, $(3, 4)$ en $(6, -1)$ zijn gegeven in het vlak. We zoeken nu het interpolatiepolynoom door deze punten. Als er twee punten zijn, kunnen we daar altijd een rechte lijn door trekken, dit is een polynoom van graad één. Bij drie punten kan altijd een parabool gevonden worden die daar doorheen gaat. In het algemeen, bij n punten kunnen we een polynoom van graad $n - 1$ vinden. Door de vier gegeven punten kunnen we dus een derdegraads polynoom vinden, dat wil zeggen een benaderende functie f van de vorm*

$$f(x) = ax^3 + bx^2 + cx + d.$$

De constanten a , b , c en d zijn nog onbekend, ons doel is om deze te bepalen. Uit de gegevens weten we dat

$$\begin{aligned} f(1) &= 3 \\ f(2) &= 2 \\ f(3) &= 4 \\ f(6) &= -1. \end{aligned}$$

Oftewel, door invullen,

$$\begin{aligned} a + b + c + d &= 3, \\ 8a + 4b + 2c + d &= 2, \\ 27a + 9b + 3c + d &= 4, \\ 216a + 36b + 6c + d &= -1. \end{aligned}$$

Oftewel, in matrixvorm opgeschreven,

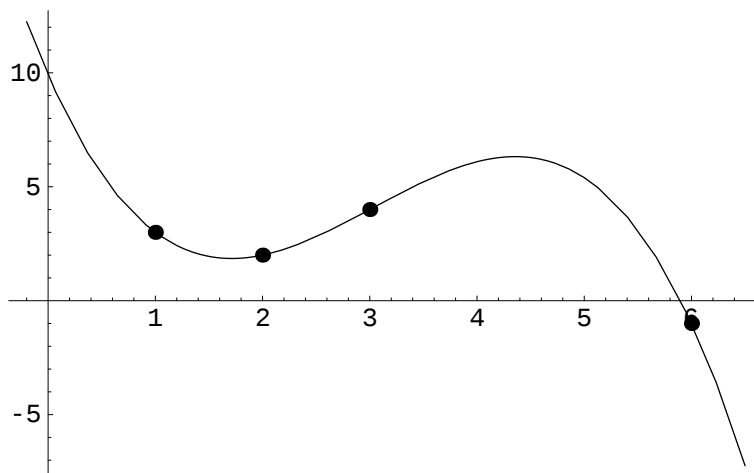
$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \\ 216 & 36 & 6 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 4 \\ -1 \end{pmatrix}.$$

Dit is een eenvoudige matrixvergelijking die we zo kunnen oplossen. De matrix die hier gebruikt wordt is van een speciale vorm en wordt de **Vandermonde-matrix** genoemd.

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \\ 216 & 36 & 6 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 3 \\ 2 \\ 4 \\ -1 \end{pmatrix} = \frac{1}{60} \begin{pmatrix} -29 \\ 264 \\ -649 \\ 594 \end{pmatrix}.$$

Dus de gezochte functie is

$$f(x) = \frac{1}{60}(-29x^3 + 264x^2 - 649x + 594).$$



FIGUUR 1. DERDEGRAADS INTERPOLATIEPOLYNOOM DOOR VIER GEGEVEN PUNTEN.

Het is nu ook duidelijk waarom je voor een n -tal gegeven punten altijd een polynoom van graad $n - 1$ kan gebruiken. Dat komt omdat in een polynoom van graad $n - 1$ het aantal vrijheidsgraden n is, namelijk één voor elke coëfficiënt voor $x^k, 0 \leq k \leq n - 1$.

De hierboven besproken methode is een algemene methode om dit soort vergelijkingen op te lossen. We hebben hierbij voor de lineaire ruimte $P_n(x)$ van polynomen in één variabele, met graad ten hoogste n , de standaardbasis gebruikt, die bestaat uit de elementen $1, x, x^2, \dots, x^n$. In de volgende paragraaf gebruiken we een andere basis voor deze ruimte en krijgen zo een andere interpolatietechniek. Deze levert uiteraard dezelfde functie op.

2.2 Lagrange-interpolatie

De standaardbasis van $P_n(x)$ heeft als nadeel dat we een matrixinversie moeten toepassen om onze functie te vinden. Nu gaan we kijken naar een voor dit doeleind meer geschikte keuze voor een basis voor $P_n(x)$. Zij $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ een $n + 1$ -tal punten in $\mathbb{R}^2$, zodanig dat geen twee x_j hetzelfde zijn. Definieer nu voor $j = 0, \dots, n$ de functies ℓ_j gegeven door

$$\ell_j(x) = \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i}.$$

Het is eenvoudig na te rekenen dat $\ell_j(x_j) = 1$ en dat $\ell_j(x_i) = 0$ voor alle $i \neq j$. Oftewel, $\ell_j(x_i) = \delta_{ij}$. Omdat de noemer de variabele x niet bevat, is ℓ_j inderdaad een polynoom voor elke j , en omdat voor elk tweetal functies geldt dat hun nulpunten niet allemaal samenvallen, zijn ze ook allemaal onderling lineair onafhankelijk, en dus vormen deze $n + 1$ polynomen een basis voor de $n + 1$ -dimensionale ruimte $P_n(x)$.

Waarom zouden we deze ingewikkeld uitziende functies als basis gebruiken? Juist vanwege de eigenschap hierboven, dat $\ell_j(x_i) = \delta_{ij}$. Definieer namelijk

$$p(x) = \sum_{i=0}^n y_i \ell_i(x).$$

Dan geldt dat $p(x_j) = y_j$ voor $j = 0, \dots, n$ en we hebben dus met een relatief eenvoudige constructie het interpolatiepolynoom te pakken.

Opmerking 2.2.1 De polynomen $\ell_j(x_i)$ worden de **Lagrange-polynomen** genoemd, het proces hierboven beschreven staat bekend als **Lagrange-interpolatie**. [5]

De vraag rijst of het handig is om in de praktijk wat mee te doen. In sommige situaties is dit inderdaad het geval, maar dan gaat het voornamelijk om ééndimensionale situaties, waarbij we de functie of formule die de punten (x_j, y_j) genereert, niet kennen of niet kunnen oplossen. Dan wordt gebruik gemaakt van een interpolatiepolynoom om tussenliggende functiewaarden te bepalen. Lagrange-interpolatiepolynomen hebben een aantal onprettige eigenschappen, waar we graag vanaf willen. Ten eerste divergeren de krommen voor $|x| \rightarrow \infty$. Ten tweede, als we alleen de punten in het vlak gegeven hebben en geen oriëntatie van het vlak, dat wil zeggen geen assenstelsel, dan ligt de kromme niet vast. Roteren van het assenstelsel geeft ons namelijk bij dezelfde punten een andere kromme. Met andere woorden, de kromme is niet **rotatie-invariant**. Ten derde werkt de methode niet als twee punten recht boven elkaar liggen. Dat zal in de uitwerking zichtbaar worden omdat de Vandermonde matrix dan singulier is, of in het Lagrange-proces twee van de x_j gelijk zijn, waardoor een deling door nul optreedt bij het bepalen van de functies ℓ_j . Uiteraard willen we krommen die voor alle verzamelingen punten werken.

2.3 Oscillaties

Er is echter nog een probleem bij het werken met deze manier van interpolatie. Aan de hand van een voorbeeld gaan we zien wat er kan gebeuren als de graad van het polynoom toeneemt.

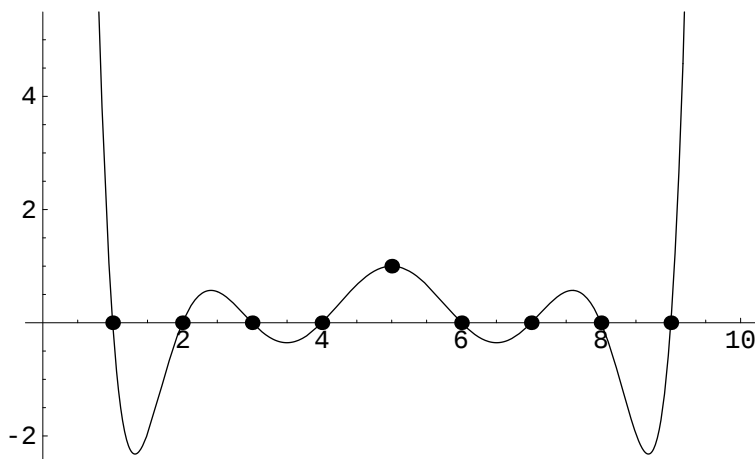
Voorbeeld 2.3.1 *Gegeven zijn de volgende voorwaarden op het platte vlak.*

$$\begin{aligned}x_i &= i \text{ voor } i = 1, \dots, 9, \\y_i &= 0 \text{ voor } i = 1, 2, 3, 4, 6, 7, 8, 9, \\y_5 &= 1.\end{aligned}$$

Alle punten, op één na, liggen op een rechte lijn. Wat we graag zouden willen is een functie die ongeveer nul is rond $i \in [1, 3]$ en $i \in [7, 9]$. We weten dat de oplossing gegeven wordt door $p(x) = \sum_{i=1}^n y_i \ell_i(x)$, in ons geval is dat $p(x) = \ell_5(x)$.

$$\begin{aligned}\ell_5(x) &= \frac{(x - x_1) \dots (x - x_4)(x - x_6) \dots (x - x_9)}{(x_5 - x_1) \dots (x_5 - x_4)(x_5 - x_6) \dots (x_5 - x_9)} \\&= \frac{1}{576}((x - 1) \dots (x - 4)(x - 6) \dots (x - 9))\end{aligned}$$

Plotten we nu een grafiek van deze functie, dan is te zien dat de grafiek allesbehalve dicht bij nul zit voor $i \leq 3$ of $i \geq 7$.



FIGUUR 2. HET PROBLEEM DAT EEN ENKEL POLYNOM VAN HOGE GRAAD KAN GEVEN

Dit is het gevolg van het feit dat we met een enkel polynoom werken en het feit dat we precies door de punten willen gaan. Een enkel polynoom kan zich niet zo goed aanpassen en gaat daarom oscilleren. Dat kan als volgt ingezien worden. Het gevonden polynoom heeft acht reële nulpunten. Gebruiken we de stelling van Rolle,

Stelling 2.3.2 (Rolle) *Tussen elke twee nulpunten van een continu differentieerbare functie f op een interval $[a, b] \in \mathbb{R}$ ligt tenminste een nulpunt van f' ,*

dan zien we dus dat de afgeleide minstens zeven nulpunten heeft. De tweede afgeleide heeft zes nulpunten, enzovoorts, de n -de afgeleide heeft minstens $8 - n$ nulpunten ($n \leq 7$). Deze heel hoge afgeleiden zijn lagegraads polynomen, die ook al naar oneindig gaan voor $|x| \rightarrow \infty$. Bijvoorbeeld, voor de in het voorbeeld gevonden f is $f^{(6)}$ gelijk aan het polynoom $35x^2 - 350x + \frac{1675}{2}$, de nulpunten liggen bij $x = 5 \pm \sqrt{\frac{210}{14}}$, dat is ongeveer bij de waarden $x = 3,96$ en $x = 6,04$. In ons geval is $f^{(6)}(9) = \frac{1045}{2}$, de zesde afgeleide heeft dus behoorlijk hoge functiewaarden. Hierdoor gaat f zich wild gedragen, ofwel **oscilleren**. Dit effect doet zich sterker voor zodra de graad van het polynoom toeneemt, een negendegraads polynoom oscilleert meestal al te veel om bruikbaar te zijn.

3 Meerdere dimensionale interpolatie

3.1 Parametrisaties

Het gebruik van een ééndimensionale functie $p(x)$ is voor het interpoleren van punten in een tweedimensionale ruimte blijkbaar niet handig. We moeten daarom overgaan op tweedimensionale functies. Bovendien is de grafiek van $p(x)$ het beeld van de tweedimensionale functie $\tilde{p}(x) = (x, p(x))$ en blijkt dus dat als we overgaan naar tweedimensionale functies, we alleen maar aan het veralgemeniseren

zijn.

We hebben dus een $n + 1$ -tal punten $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ gegeven, dat we graag willen interpoleren. We doen dat op de volgende manier. We voegen op elk punt een derde coördinaat toe, die gelijk is aan de index van dat punt, we verkrijgen dan punten van de vorm $p_j = (j, x_j, y_j) (j = 0, \dots, n)$. We gebruiken nu de extra coördinaat als “loopvariabele”, die we t noemen. We definiëren een tweedimensionale functie $f(t) = (x(t), y(t))$, waarbij $x(t)$ het interpolatiepolynoom van de punten (j, x_j) is en $y(t)$ die van de punten (j, y_j) . Als we van deze functie f de beeldkromme c in $\mathbb{R}^2$ bekijken voor $t \in [0, n]$, dan zien we dat $x(j) = x_j$ en $y(j) = y_j$. Het gevolg is dat de kromme c door de punten (x_j, y_j) gaat, met (x_0, y_0) als beginpunt en (x_n, y_n) .

Wat we krijgen is een tweedimensionaal interpolatiepolynoom. We kunnen dit uiteraard in elke dimensie k uitvoeren om zodoende een k -dimensionaal interpolatiepolynoom te verkrijgen van $n + 1$ punten $p_0, p_1, \dots, p_n \in \mathbb{R}^k$. Normaliter wordt in de praktijk in dimensies 2 en 3 gewerkt.

3.2 Approximatie

We hebben gezien in voorbeeld 2.3.1 dat eisen dat we precies door de punten gaan, tot vervelende resultaten kan leiden. Het lijkt dus een goed idee om deze eis te laten vallen. We krijgen dan te maken met een heel ander onderwerp dan interpolatie, namelijk **approximatie**. We gaan de gegeven punten benaderen met een gladde kromme inplaats van er precies doorheen te gaan, en kijken wat dat oplevert.

Om oscillaties te voorkomen, gaan we het probleem anders aanpakken. Om een benaderende (approximerende) kromme te verkrijgen voor de gegeven punten, gaan we een soort gewogen gemiddelden nemen van deze punten, afhankelijk van een soort tijdsvariabele $t \in [0, 1]$, zodanig dat als $t = 0$, we in het eerste punt zijn, en voor $t = 1$ zitten we in het laatste punt. De weegfactoren zijn dus functies van t , gedefinieerd op het interval $[0, 1]$. Eerst definieer ik het begrip “gewogen gemiddelde” van een verzameling punten wat nader.

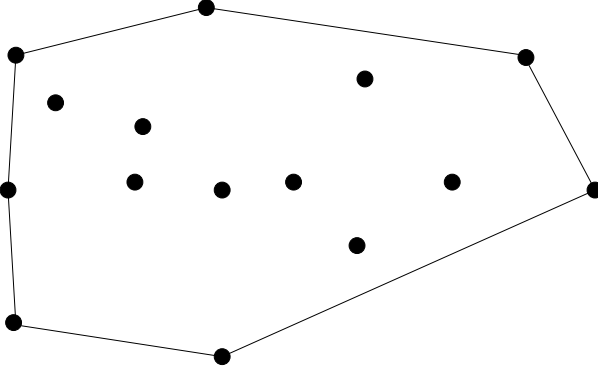
Definitie 3.2.1 Een **convexe combinatie** van $n + 1$ punten $p_0, \dots, p_n$ is een lineaire combinatie $\lambda_0 p_0 + \dots + \lambda_n p_n$ die voldoet aan $\forall i \lambda_i \geq 0$ en $\sum_{i=0}^n \lambda_i = 1$.

Bedenk dat dit inderdaad veralgemeniseert wat je van gemiddelde gewend bent bij getallen. De λ_i zijn de **gewichten**, en we nemen elke p_i met een weegfactor λ_i , zodanig dat de som van de weegfactoren 1 is. Het gewone gemiddelde van n getallen gebruikt bijvoorbeeld $\forall i \lambda_i = 1/n$.

Als we het gemiddelde van getallen of punten nemen, blijven we altijd “ergens daar tussenin”. Ook dat laatste definiëren we preciezer.

Definitie 3.2.2 De *convex omhullende* van een verzameling punten X is de doorsnede van alle convexe verzamelingen die X bevatten.

Als je een verzameling punten p_i gegeven hebt, blijkt het zo te zijn dat enerzijds, alle convexe combinaties inderdaad “er tussenin” liggen en anderzijds dat alle punten die “er tussenin” liggen één of andere convexe combinatie zijn. De bewering is dat de verzameling mogelijke gewogen gemiddelden van een gegeven verzameling punten precies de convex omhullende van deze verzameling is. We houden het hier bij eindige verzamelingen punten.



FIGUUR 3. DE CONVEX OMHULLENDE VAN EEN EINDIGE VERZAMELING PUNTEN

Stelling 3.2.3 Zij X een verzameling punten met $|X| = n < \infty$ en zij C de convex omhullende van X . Zij C' de verzameling convexe combinaties van punten in X . Dan is $C = C'$.

Bewijs. Eerst laten we zien dat C' convex is. Zij $a, b \in C'$. Dan is $a = \lambda_0 p_0 + \dots + \lambda_n p_n$ en $b = \mu_0 p_0 + \dots + \mu_n p_n$ met $\lambda_0, \dots, \lambda_n, \mu_0, \dots, \mu_n \geq 0$ en $\sum_{i=0}^n \lambda_i = 1 = \sum_{i=0}^n \mu_i$. Laat nu $\lambda, \mu \geq 0$ zodat $\lambda + \mu = 1$, dan is $\lambda a + \mu b = \lambda(\lambda_0 p_0 + \dots + \lambda_n p_n) + \mu(\mu_0 p_0 + \dots + \mu_n p_n) = (\lambda \lambda_0 + \mu \mu_0) p_0 + \dots + (\lambda \lambda_n + \mu \mu_n) p_n$. Waarbij $(\lambda \lambda_0 + \mu \mu_0) + \dots + (\lambda \lambda_n + \mu \mu_n) = \lambda(\lambda_0 + \dots + \lambda_n) + \mu(\mu_0 + \dots + \mu_n) = \lambda + \mu = 1$. Dus $\lambda a + \mu b$ is een convexe combinatie van punten in X en ligt dus in C' . Omdat a, b en λ willekeurig waren, is C' dus convex. Omdat C de doorsnede is van alle convexe verzamelingen, is $C \subseteq C'$.

Rest nog te bewijzen dat $C' \subseteq C$, oftewel dat elke convexe combinatie van punten in X gelegen is in C , oftewel gelegen is in alle convexe verzamelingen die X bevatten.

Laat $a \in C'$. Dan $a = \lambda_0 p_0 + \dots + \lambda_n p_n$ met alle $\lambda_i \geq 0$. Punten met coëfficiënt nul laten we weg. Dan is $a = \lambda_0 p_0 + (1 - \lambda_0)(\frac{\lambda_1}{1 - \lambda_0} p_1 + \dots + \frac{\lambda_n}{1 - \lambda_0} p_n)$. Omdat $\lambda_1 + \dots + \lambda_n = 1 - \lambda_0$, oftewel $\frac{\lambda_1}{1 - \lambda_0} + \dots + \frac{\lambda_n}{1 - \lambda_0} = 1$, is de tweede term een convexe combinatie a_n van punten in X met een term minder dan a_n , en ligt a op de bindingslijn tussen p_0 en a_n . Ofwel, als $a_n \in C$, dan ook $a \in C$. Op soortgelijke

manier kunnen we uit a_n een a_{n-1} construeren, die slechts $n - 1$ termen heeft. Zo gaan we door tot a_2 , die nog twee termen heeft en dus op de verbindingslijn tussen p_{n-1} en p_n ligt. Dus $a_2 \in C$, en terugredenerend langs $a_3, \dots, a_n$ concluderen we dat $a \in C$. Enige aanname was $a \in C'$, dus de conclusie is $C' \subseteq C$. Dus $C' = C$. $\square$

De reden dat we hier aandacht aan besteden is om een eigenschap van de krommen die we in het volgende hoofdstuk introduceren, convexe combinaties van hun definiërende punten zijn, en we daarom meteen kunnen concluderen dat ze binnen de convex omhullende van deze punten blijven, een eigenschap die ook wel de **convexe-omhullingseigenschap** wordt genoemd.

4 Bézierkrommen

We willen dus, gegeven een aantal punten een kromme definiëren die langs (niet noodzakelijk door) deze punten gaat en binnen de convex omhullende blijft. We gaan een parametrische kromme maken, oftewel een die afhangt van een soort tijdsvariabele $t \in [0, 1]$. We zorgen ervoor dat voor $t = 0$ we in het beginpunt zitten en voor $t = 1$ in het eindpunt. We bekijken eerst het eenvoudigste geval.

4.1 Twee punten

Twee punten, x_0 en x_1 , zijn gegeven in een n -dimensionale ruimte (het zijn dus n -dimensionale vectoren). De convex omhullende van twee punten is slechts een lijnstuk. De enige kromme die voldoet is dat lijnstuk. De kromme is dezelfde als de lineaire interpolatiekromme van deze twee punten:

$$P_{01}(t) = (1 - t)x_0 + tx_1$$

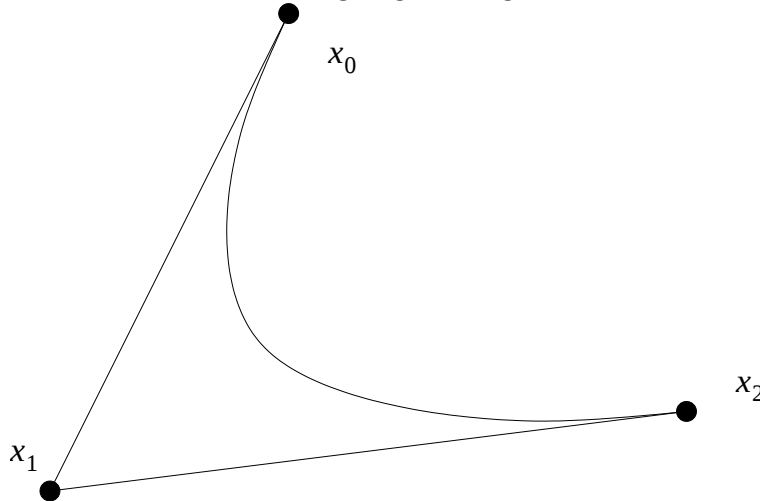
Dit is een polynoom van graad één. Ik heb de kromme hier P_{01} genoemd omdat deze de punten x_0 en x_1 interpoleert. We kunnen zo van twee opeenvolgende punten een benaderingskromme construeren.

4.2 Drie punten

We kijken naar drie punten, x_0 , x_1 en x_2 en willen de benaderende kromme maken voor deze punten. We doen dat door het bovenstaande principe herhaald toe te passen:

$$\begin{aligned} P_{01}(t) &= (1 - t)x_0 + tx_1 \\ P_{12}(t) &= (1 - t)x_1 + tx_2 \\ P_{02}(t) &= (1 - t)P_{01}(x) + tP_{12}(x) \end{aligned}$$

Het polynoom $P_{02}(t)$ is er nu een van graad twee. We krijgen nu een kromme die door de punten x_0 en x_2 gaat, maar in het algemeen níet door het punt x_1 . Wel is het zo dat de kromme zogezegd “aangetrokken” wordt door het punt x_1 .



FIGUUR 4. TWEDEGRAADS BÉZIERKROMME.

4.3 Het algemene geval

We bekijken $m+1$ punten $x_0, \dots, x_m \in \mathbb{R}^n$. Deze punten worden de **stuurpunten** genoemd, aangezien ze de kromme sturen. We definiëren de polynomen $P_{ij}(t)$ recursief door

$$\begin{aligned} P_{ii}(t) &= x_i && \text{voor } i = 0, \dots, m; \\ P_{ij}(t) &= (1-t)P_{i,j-1}(x) + tP_{i+1,j}(x) && \text{voor } i < j. \end{aligned}$$

Informeel gezegd nemen we dus gemiddelden van gemiddelden. Daardoor nemen we steeds convexe combinaties van stuurpunten, en kunnen we concluderen dat de kromme altijd binnen de convexe omhullende van zijn stuurpunten blijft. Het is opmerkenswaardig dat het polynoom P_{ij} altijd van graad $j-i$ is, ook in het geval $i=j$ want dan is P_{ij} een constant polynoom.

Definitie 4.3.1 *Het polynoom P_{0m} wordt de **Bézierkromme** voor de punten $x_0, \dots, x_m$ genoemd.*

Opmerking 4.3.2 *Het hier toegepaste principe van gemiddelden van gemiddelden nemen is de grondslag voor de Bézierkromme en wordt het **de Casteljau-algoritme** genoemd. Dit algoritme is vernoemd naar de Franse ingenieur Paul de Casteljau, die werkzaam was bij autofabrikant Citroën. De krommen zelf zijn vernoemd naar de eveneens Franse ingenieur Pierre Bézier. Bézier was werkzaam bij Renault, alwaar hij deze krommen gebruikte voor het ontwerp van carrosserieën. [1]*

Het nemen van gemiddelden kun je zien als een functie van twee punten. Het construeren van een Bézierkromme met het de Casteljau-algoritme is het herhaaldelijk toepassen van deze functie, die resulteert in een formule voor de Bézierkromme die we in de volgende paragraaf afleiden. Eerst kijken we naar een reeks eigenschappen van Bézierkrommen die prettig zijn voor gebruik in de praktijk.

Opmerking 4.3.3 (Eigenschappen van Bézierkrommen) *Bézierkrommen voldoen aan een aantal eigenschappen:*

- **Partitie van de eenheid:** *de coëfficiëntfuncties voor de steunpunten tellen op tot de functie die identiek één is. Daardoor zijn de punten op de kromme altijd convexe combinaties van de steunpunten.*
- **de convexe omhullingseigenschap:** *de kromme ligt geheel binnen de convex omhullende van de steunpunten. Dit is een gevolg van de vorige eigenschap.*
- **de variatieverminderende eigenschap:** *voor een Bézierkromme in het vlak geldt dat geen enkele lijn de kromme vaker snijdt dan zijn steunpolygoon. Met het **steunpolygoon** van een Bézierkromme bedoelen we de verzameling verbindingslijnen tussen opeenvolgende steunpunten voor deze kromme.*
- **Affiene invariantie.** *Het toepassen van een affine transformatie Φ op de punten van de Bézierkromme heeft hetzelfde effect als het toepassen van Φ op de steunpunten en het vervolgens construeren van de kromme op de nieuwe punten [4]. Een **affiene transformatie** is een lineaire transformatie gevolgd door een translatie: $\Phi(x) = Ax + b$, met A een transformatiematrix en b een translatievector.*

In de volgende paragraaf gaan we kijken naar de formule die de Bézierkromme definieert.

4.4 De afleiding van de formule

We gaan kijken wat er gebeurt als we het de Casteljau-algoritme toepassen op een $m + 1$ -tal gegeven punten $x_0, \dots, x_m \in \mathbb{R}^n$. Omdat we alleen maar convexe combinaties nemen van convexe combinaties, weten we dat het eindresultaat een convexe combinatie is. Om een idee te krijgen, kijken we eerst wat er gebeurt met een beperkt aantal punten.

Voorbeeld 4.4.1 De punten x_0, x_1, x_2 en x_3 zijn gegeven en we zoeken voor deze vier punten de Bézierkromme $B(t) = P_{03}(t)$.

$$\begin{aligned}
P_{01}(t) &= (1-t)x_0 + tx_1 \\
P_{12}(t) &= (1-t)x_1 + tx_2 \\
P_{23}(t) &= (1-t)x_2 + tx_3 \\
P_{02}(t) &= (1-t)P_{01}(t) + tP_{12}(t) \\
&= (1-t)^2x_0 + 2t(1-t)x_1 + t^2x_2 \\
P_{13}(t) &= (1-t)P_{12}(t) + tP_{23}(t) \\
&= (1-t)^2x_1 + 2t(1-t)x_2 + t^2x_3 \\
P_{03}(t) &= (1-t)P_{02}(t) + tP_{13}(t) \\
&= (1-t)^3x_0 + 3t(1-t)^2x_1 + 3t^2(1-t)x_2 + t^3x_3
\end{aligned}$$

Wat opvalt in de laatste formule is dat de machten van t en de indices van de x in elke term overeenkomen. Dat komt omdat in de recursieve formule voor de P_{ij} de P_{i+1j} met een voorfactor $1-t$ komt en de P_{ij-1} met een voorfactor t . Dat is ook de reden dat de som van de exponenten van t en $1-t$ in elke term optellen tot 3, de graad van het polynoom. Iets anders dat opvalt zijn de voorfactoren 1, 2, 1 in P_{02} en P_{13} en de voorfactoren 1, 3, 3, 1 uit P_{03} , die we ook in de driehoek van Pascal tegenkomen.

Werken we de som van de voorfactoren uit, dan zien we het volgende gebeuren:

$$\begin{aligned}
&(1-t)^3 + 3t(1-t)^2 + 3t^2(1-t) + t^3 \\
&= (-t^3 + 3t^2 - 3t + 1) + 3t(t^2 - 2t + 1) + 3t^2(-t + 1) + t^3 \\
&= -t^3 + 3t^2 - 3t + 1 + 3t^3 - 6t^2 + 3t - 3t^3 + 3t^2 + t^3 \\
&= 1
\end{aligned}$$

We concluderen dat voor elke $t \in [0, 1]$ het polynoom $P_{03}(t)$ een convexe combinatie is. We hadden dat ook direct kunnen inzien met behulp van het **Binomium van Newton**, dat zegt dat

$$(a+b)^r = \sum_{k=0}^r \binom{r}{k} a^k b^{r-k}$$

en dus voor $a = t$, $b = 1-t$ dat

$$1 = (t + 1 - t)^r = \sum_{k=0}^r \binom{r}{k} t^k (1-t)^{r-k}.$$

Deze aanpak, ook gebruikt in [4], geeft direct aanleiding tot het algemeen definiëren van de Bézierkromme op de volgende manier.

Definitie 4.4.2 (Bézierkromme) Voor $m+1$ gegeven stuurpunten $x_0, \dots, x_m \in \mathbb{R}^n$ is de Bézierkromme voor deze punten gegeven door de formule

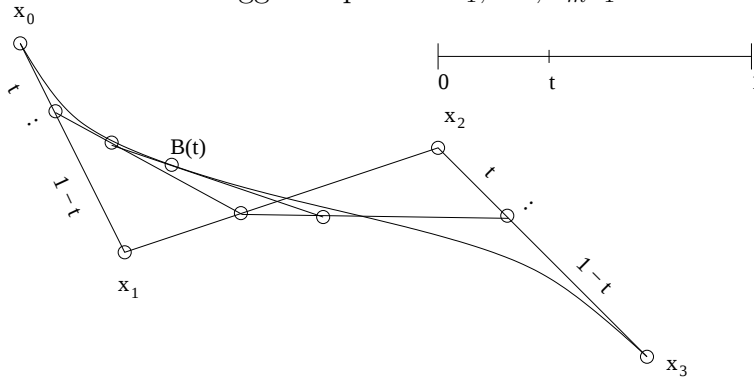
$$B(t) = \sum_{k=0}^m B_k^m(t)x_k,$$

waarbij

$$B_k^m(t) = \binom{m}{k} t^k (1-t)^{m-k}.$$

De functies B_k^m doen dienst als de weegfactoren in het gewogen gemiddelde en worden de **Bernstein-polynomen** genoemd. Deze formule voor de Bézier-kromme heet ook wel de **Bézier-Bernstein representatie**.

Uit de definitie volgt direct dat $B(t)$ een convexe combinatie van de punten $x_0, \dots, x_m$ is voor alle $t \in [0, 1]$: de functies B_k^m optellen tot de functie die identiek één is. Ook volgt direct dat $B(0) = x_0$ en $B(1) = x_m$. De kromme gaat doorgaans niet door de tussenliggende punten $x_1, \dots, x_{m-1}$.



FIGUUR 5. HET DE CASTELJAU-ALGORITME.

Uit Figuur 5 blijkt dat de kromme zich “naar de tussenpunten toe trekt”. Vanwege deze eigenschap en omdat we binnen een beperkt gebied blijven, hebben we van oscillaties geen last. Er is wel een ander probleem dat zich blijft voordoen. Dat heeft te maken met het gebrek aan controle over de vorm van de kromme. Als we een Bézierkromme van hoge graad, zeg 10, hebben, en we verplaatsen één punt, dan verandert de hele kromme. Als we dus een gegeven kromme willen benaderen en dat lukt overal behalve op één bepaalde plaats, en we proberen dat op die plaats te verhelpen door een stuurpunt x_k te verplaatsen, dan verandert alles mee en moeten we alle andere punten ook ietsje verplaatsen. De oorzaak van dit feit is dat de functie B_k^m niet nul is op het hele open interval $(0, 1)$ en de betreffende term in de som op dat hele interval van waarde verandert als x_k wordt verplaatst. We kunnen dit verhelpen door meerdere Bézierkrommen van lagere graad aan elkaar te plakken, en vervolgens de punten zó te plaatsen dat de eindjes vloeiend in elkaar overlopen. Ondanks dat het een eenvoudig idee lijkt, is het een heel belangrijke stap geweest, want het is precies dit idee dat aan de **spline** ten grondslag ligt!

4.5 Afgeleiden van Bézierkrommen in de eindpunten

Vanaf dit punt gaan we ons specifiek richten op Bézierkrommen van graad drie, gegeven door vier verschillende punten $x_0, \dots, x_3$. Of we in de $\mathbb{R}^2$ of $\mathbb{R}^3$ werken, laten we in het midden. We hebben dus twee eindpunten x_0 en x_3 en twee tussenpunten x_1 en x_2 . Wat we later gaan doen, is meerdere van deze krommen aaneensluiten in de eindpunten, en we willen graag dat dat vloeiend verloopt. Daarom hebben we informatie nodig over wat de afgeleiden doen in de eindpunten van een Bézierkromme. Daarom gaan we daar nu naar kijken.

Stelling 4.5.1 *Een Bézierkromme raakt in elk eindpunt aan het lijnstuk dat dat eindpunt verbindt met het daar direct aan gelegen tussenpunt. Daarmee wordt het volgende bedoeld. Het lijnstuk dat x_0 verbindt met x_1 raakt in x_0 aan de Bézierkromme, alwaar $t = 0$, en het lijnstuk dat x_m verbindt met x_{m-1} raakt in x_m aan de Bézierkromme in het punt x_m , dat is voor $t = 1$.*

Dat is niet moeilijk in te zien. Hier volgen de formules voor derdegraads Bézierkrommen. De afgeleide van het Bézierpolynoom is gelijk aan

$$\begin{aligned} B'(t) &= \frac{d}{dt}(1-t)^3x_0 + 3t(1-t)^2x_1 + 3t^2(1-t)x_2 + t^3x_3 \\ &= -3(1-t)^2x_0 + (3(1-t)^2 - 6t(1-t))x_1 + (-3t^2 + 6t(1-t))x_2 \\ &\quad + 3t^2x_3. \end{aligned}$$

Invullen voor $x = 0$ en $x = 1$ geeft

$$\begin{aligned} B'(0) &= -3x_0 + 3x_1, \\ B'(1) &= -3x_2 + 3x_3. \end{aligned}$$

Omdat alle termen in de Bézier-Bernstein representatie een factor t^2 hebben, behalve twee stuks, hebben al deze termen in de eerste afgeleide een factor t . Die vervallen als we $t = 0$ invullen. Hetzelfde geldt voor $t = 1$. Alle termen hebben een factor $(1-t)^2$, behalve twee stuks, en slechts die twee blijven over als we na het differentiëren $t = 1$ invullen. Ditzelfde zal werken voor hogere afgeleiden, maar telkens als de orde van differentiatie toeneemt, neemt het aantal termen dat vervalt af: steeds meer machten van t en $1-t$ worden weggedifferentieerd.

Je ziet dat de afgeleide in $x = 0$ gelijk is aan de vector die van x_0 naar x_1 wijst en als lengte drie keer de afstand tussen deze punten heeft. In het eindpunt gebeurt hetzelfde; de afgeleide vector wijst van x_2 naar x_3 en heeft als lengte drie keer de afstand tussen deze punten. Voor de tweede afgeleide van een derdegraads Bézierkromme geldt

$$\begin{aligned} B''(t) &= \frac{d}{dt}[-3(1-t)^2x_0 + (3(1-t)^2 - 6t(1-t))x_1 \\ &\quad + (-3t^2 + 6t(1-t))x_2 + 3t^2x_3] \\ &= 6(1-t)x_0 + 6tx_1 - 12(1-t)x_1 - 12tx_2 + 6(1-t)x_2 + 6tx_3 \\ &= (6x_0 - 12x_1 + 6x_2)(1-t) + (6x_1 - 12x_2 + 6x_3)t. \end{aligned}$$

Invullen geeft nu dat

$$\begin{aligned} B''(0) &= 6(x_0 - 2x_1 + x_2), \\ B''(1) &= 6(x_1 - 2x_2 + x_3). \end{aligned}$$

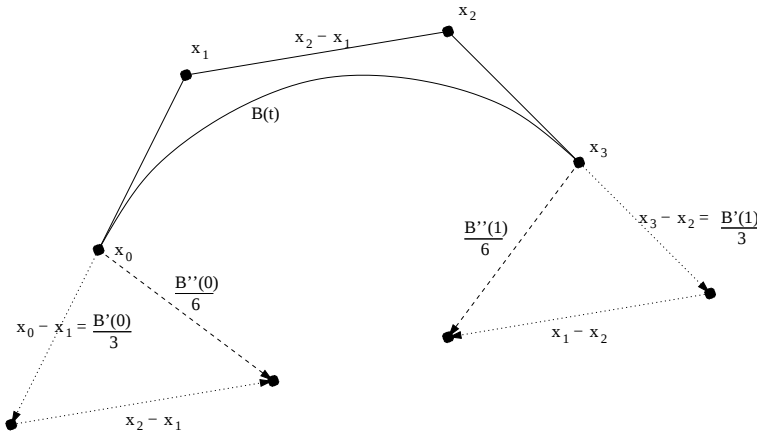
We kunnen de afgeleiden in de eindpunten illustreren als vectoren die beginnen bij het corresponderende eindpunt van de kromme. De vector $b - a$ is de vector b , die van 0 naar b loopt, opgeteld bij de vector $-a$, die van a naar 0 loopt. De vector $b - a$ loopt dus van a naar b .

De eerste afgeleide in het beginpunt, $B'(0) = 3(x_1 - x_0)$ is de vector van x_0 naar x_1 , drie keer verlengd.

De eerste afgeleide in het eindpunt, $B'(1) = 3(x_3 - x_2)$ is de vector van x_2 naar x_3 , drie keer verlengd.

De tweede afgeleide in het beginpunt, $B''(0) = 6(x_0 - 2x_1 + x_2)$ is het zesvoud van de som van de vectoren $x_0 - x_1$ en $x_2 - x_1$. Dit is de manier zoals we deze kunnen construeren. In Figuur 6 is dat gedaan.

Met de tweede afgeleide in het eindpunt, $B''(1) = 6(x_1 - 2x_2 + x_3)$, doen we hetzelfde: we schrijven deze als zes keer de som van twee vectoren, $x_3 - x_2$ en $x_1 - x_2$. Zo krijgen we voor een willekeurige kromme het volgende plaatje.



FIGUUR 6. AFGELEIDEN VAN EEN DERDEGRAADS BÉZIERKROMME IN DE EINDPUNTEN.

De afgeleidevectoren zijn hier geschaald met factoren drie en zes voor de vectoren voor de eerste respectievelijk de tweede afgeleide. Bedenk dat de derde afgeleide een constante vector is: omdat alle machten van t hoogstens drie zijn worden ze volledig weggedifferentieerd na de derde differentiatiestap en houden we een waarde over die alleen afhangt van de vier punten x_0, x_1, x_2 en x_3 .

Nu we iets over de afgeleiden weten, kunnen we Bézierkrommen aan elkaar gaan aansluiten in de eindpunten, zodanig dat afgeleiden van zekere graad overeenkomen. Anders gezegd, we kunnen nu een Bézierspline maken. Dit wordt ook wel een **samengestelde Bézierkromme** genoemd.

5 Samengestelde Bézierkrommen

We bekijken twee Bézierkrommen, B en C . Kromme B is gedefinieerd op de punten x_0, x_1, x_2 en x_3 de kromme C op y_0, y_1, y_2 en y_3 . We gaan het eindpunt van B met het beginpunt van C verbinden:

$$B(1) = C(0) \implies x_3 = y_0.$$

Nu is de samengestelde kromme continu, ofwel C^0 -differentieerbaar, in het verbindingspunt. Bedenk dat de krommen zelf altijd C^∞ -differentieerbaar zijn. Voor het gemak hernoemen we de punten van C door x_3, x_4, x_5 en x_6 , in de logische volgorde. We gaan kijken naar de eerste afgeleide voor een voldoende voorwaarde voor C^1 -differentieerbaarheid.

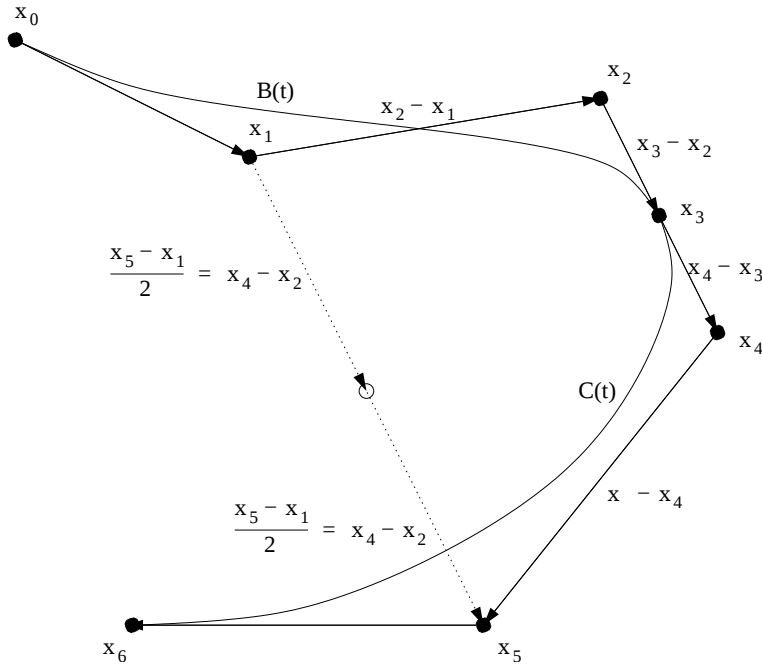
$$\begin{aligned} B'(1) = C'(0) &\iff 3(x_3 - x_2) = 3(x_4 - x_3), \\ &\iff 2x_3 = (x_2 + x_4). \end{aligned}$$

In de onderste regel zien we dat het verbindingspunt het gemiddelde moet zijn van x_2 en x_4 . Anders gezegd, x_2 en x_4 zijn elkaars puntgespiegelde ten opzichte van het schakelpunt x_3 . Gegeven de een, ligt de ander vast. Anders gezegd, de punten x_2, x_3 en x_4 zijn collineair, met x_3 precies in het midden tussen x_2 en x_4 . Dit is ook intuïtief duidelijk: om de kromme C^1 -differentieerbaar te laten zijn, moeten de raaklijnen overeenkomen en we hebben gezien dat de raaklijnen de verbindingslijnen tussen de gegeven stuurpunten zijn.

Laten we nu kijken naar de tweede afgeleide en wat we moeten doen om de aansluiting ook C^2 -differentieerbaar te krijgen.

$$\begin{aligned} B''(1) = C''(0) &\iff 6(x_3 - 2x_2 + x_1) = 6(x_5 - 2x_4 + x_3), \\ &\iff x_1 - 2x_2 + 2x_4 - x_5 = 0, \\ &\iff x_5 - x_1 = 2(x_4 - x_2). \end{aligned}$$

De formule is zodanig gemanipuleerd dat deze eenvoudig meetkundig te interpreteren is: de formule op de laatste regel zegt dat de vector van x_1 naar x_5 in dezelfde richting moet lopen als die van x_2 naar x_4 en twee keer zo lang moet zijn. In Figuur 7 is dat geïllustreerd.



FIGUUR 7. SAMENGESTELDE BÉZIERKROMME VAN GRAAD 3, DIE C^2 -DIFFERENTIEERBAAR AANSLUIT IN HET VERBINDINGSPUNT.

Merk overigens op dat bij de tweede implicatie gebruik is gemaakt van de C^1 -differentieerbaarheidseigenschap die we al hadden.

Een C^3 -differentieerbaarheid kunnen we in theorie realiseren, maar dat betekent, dat gegeven x_0 , ook x_6 vast ligt, en dat betekent dat we gegeven de kromme B , we niets meer te kiezen hebben en dat kromme C in zijn geheel vastligt. Dat is ook logisch, want twee stuksgewijs gedefinieerde derdegraads polynomen kunnen alleen maar op een C^3 -differentieerbare manier aan elkaar verbonden worden als ze door dezelfde formule worden gegeven. Zou men behoefte hebben aan een hogere graad van differentieerbaarheid, dan moet men een hogeregraads kromme gaan gebruiken. In de praktijk is C^2 -differentieerbaarheid vaak voldoende en worden derdegraads Bézierkrommen gebruikt.

6 B-splines

6.1 B-Splines van graad 3

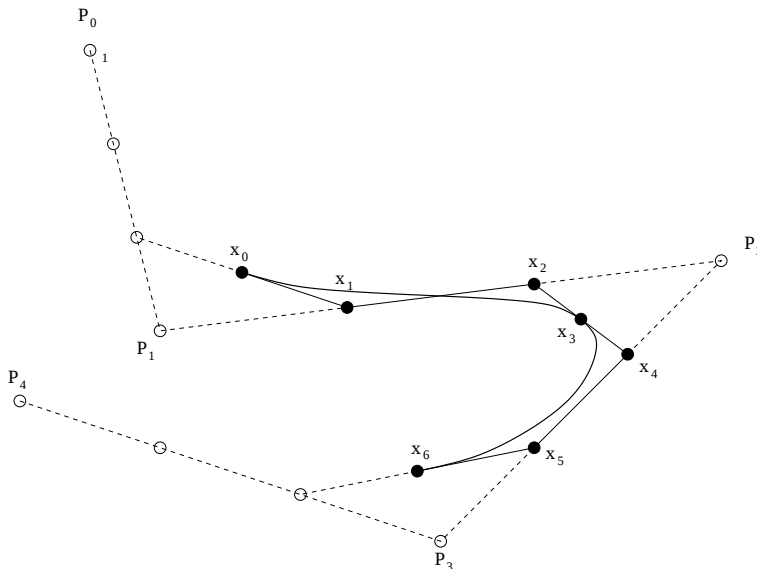
We bouwen voort op het voorgaande; we gaan nu derdegraads Bézierkrommen aaneenschakelen tot splines. De resulterende spline wordt ook wel **B-spline** genoemd. Wat we weten van Bézierkrommen is hoe we ze C^2 -differentieerbaar op elkaar aansluiten: het schakelpunt ligt midden tussen de twee direct aangrenzende stuurpunten en de vector tussen de daar weer aan liggende stuurpunten wijst in dezelfde richting en is twee keer zo lang. Kijk nogmaals naar Figuur 7.

Het feit dat de vector van x_1 naar x_5 twee keer die van x_2 naar x_4 is, betekent dat als ik x_1 puntspiegel in x_2 en x_5 puntspiegel in x_4 , ik in hetzelfde punt aankom. Laten we dat punt even P_2 noemen. Dat punt ligt een eindje van de kromme vandaan, maar als je de andere punten wegdenkt gedraagt het zich ongeveer als controlepunt: de kromme wordt er naar toe getrokken.

Bedenk dat we hier begonnen zijn met de punten voor de Bézierkromme en daar voorwaarden aan stelden. Bij elke kromme die we toevoegen, voegen we drie punten toe maar twee ervan liggen al vast door de C^2 -differentieerbaarheids-eigenschap.

Dat willen we eigenlijk niet: we willen totale vrijheid in de punten. Dat doen we op de volgende manier.

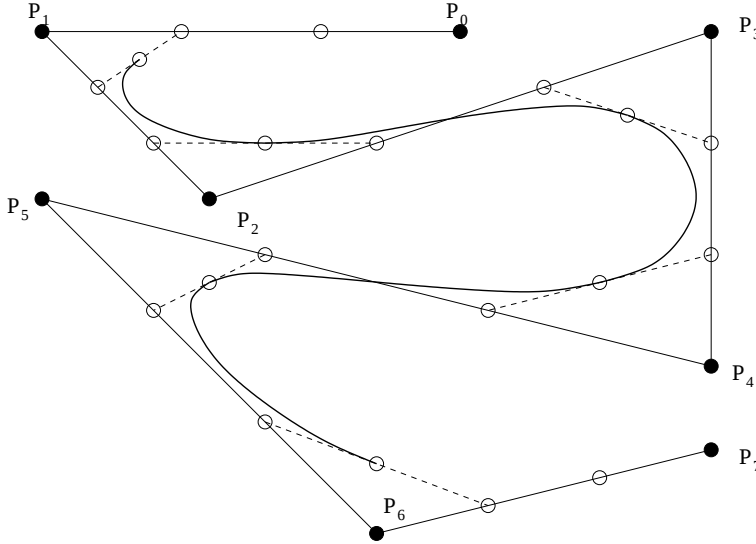
We voeren de puntspiegeling die we net in de buurt van x_3 hebben uitgevoerd, ook uit bij x_0 en x_6 : we construeren P_1 bij x_0 door x_2 te puntspiegelen om x_1 en puntspiegelen x_4 om x_5 om P_3 te verkrijgen. Als je dat doet, krijgen we twee viertallen collineaire punten: P_1, x_1, x_2, P_2 zijn collineair en op gelijke afstand van elkaar en P_2, x_4, x_5, P_3 ook. Merk nu het volgende op: Als alleen de punten P_1, P_2 en P_3 gegeven zijn, kunnen we x_1, x_2, x_4 en x_5 daaruit construeren! Bovendien weten we dat x_3 midden tussen x_2 en x_4 ligt, dus die is ook te construeren. Eindpunten x_0 en x_6 zijn nog niet te construeren, maar dat lossen we op door ook een P_0 en een P_4 te kiezen zodanig dat x_0 en x_6 op de goede plaats liggen.



FIGUUR 8. B-SPLINE-PUNTEN AFGELEID UIT DE BÉZIER-PUNTEN VORMEN EEN B-SPLINE VAN GRAAD DRIE.

In plaats van dat we de punten x_0, x_1, x_2 en x_3 opgeven voor de eerste kromme en vervolgens x_6 voor de tweede kromme (totaal vijf punten), hebben we nu als

datapunten de punten $P_0, \dots, P_4$ (ook vijf punten). Bovendien is het geheel nu volledig symmetrisch in de zin dat niet de ene kromme meer vrijheid heeft dan de andere. Bovendien is dit systeem eenvoudig uit te breiden: elk punt P_i extra, ongeacht aan welke kant, levert een extra Bézierkromme, die automatisch C^2 -differentieerbaar aansluit. Voor een wat langere spline, zie Figuur 9.



FIGUUR 9. EEN VOORBEELD VAN EEN B-SPLINE VAN GRAAD DRIE.

6.2 De formule voor de B-spline

We zullen nu de B-spline exact definiëren zoals in [4]. We kiezen een **graad** n voor de spline. In de praktijk wordt meestal $n = 3$ aangehouden, omdat $n \leq 2$ te beperkte mogelijkheden heeft en $n \geq 4$ onnodig ingewikkeld is. Vervolgens definiëren we de **basisfuncties**. Die gelden voor alle B-splines en hangen daarom nog niet af van de stuurpunten.

Definitie 6.2.1 (B-spline basisfuncties) Zij $0 = u_0, u_1, \dots, u_N = 1$ een niet-dalende rij getallen binnen $[0, 1]$, dit worden de **knopen** van de spline genoemd. Zij $P_0, \dots, P_n$ een $n+1$ -tal gegeven punten, met $n > m$. Dit zijn de **stuurpunten**. Kies een getal n als **graad** van de spline. De **orde** van een spline is altijd één hoger dan de graad. Definieer de basisfuncties van orde 1 als volgt:

$$B_{i,1} = \mathbf{1}_{[u_i, u_{i+1})},$$

$$B_{N-1,1} = \mathbf{1}_{[u_{N-1}, u_N]}$$

en definieer de functies van hogere orde met de recursieve formule

$$B_{i,k}(t) = \frac{t - u_i}{u_{i+k-1} - u_i} B_{i,k-1}(t) + \frac{u_{i+k} - t}{u_{i+k} - u_{i+1}} B_{i+1,k-1}(t),$$

In deze formule loopt k van 2 naar $n + 1$ over de ordes van de basisfuncties. De hoogste orde is dus de orde van de uiteindelijke spline. De parameter i loopt van 0 naar $N - k$, dit geeft de deelintervallen aan.

Bovenstaande definitie is gelijkwaardig aan die in [6]. Let hier wel op, sommige literatuur gebruikt als onderindex niet de orde, maar de graad van de basisfuncties. De formules gaan er dan iets anders uitzien.

Deze basisfuncties hangen af van de knopen die gekozen zijn. Na het geven van de **stuurpunten** kan men nu de B-spline-kromme vastleggen.

Definitie 6.2.2 (B-spline) *Zij $P_{-n}, \dots, P_{N-1}$ een $N + n$ -tal stuurpunten. De **B-spline** met orde $n + 1$ voor de stuurpunten P_i en de knopen u_i wordt gegeven door*

$$\mathbf{x}(u) = \sum_{i=-n}^{N-1} P_i B_{i,n+1}(u).$$

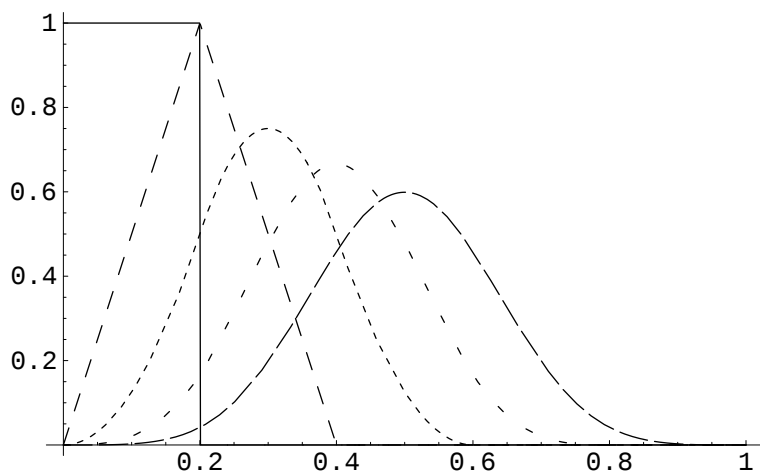
Net als bij Bézierkrommen zijn deze splines afhankelijk van de stuurpunten, ze werken ongeveer op dezelfde manier. We gaan nu kijken naar de voordelen van B-splines boven Bézierkrommen en kijken eerst naar de verschillen wat betreft hun gladheid.

6.3 Differentieerbaarheidseigenschappen van B-splines

Bij het samenstellen van Bézierkrommen was het nodig om de punten op de goede plaats te leggen om aan bepaalde differentieerbaarheidseigenschappen te voldoen. Dat is bij B-splines niet meer het geval. Omdat alle basisfuncties al differentieerbaar zijn, is de kromme dat automatisch ook.

De basisfuncties zijn C^{n-1} -differentieerbaar in de knopen. Het is mogelijk om meerdere knopen op dezelfde waarde te leggen, men spreekt dan van **meervoudige knopen**. Als q knopen over elkaar liggen, dan spreekt men van een knoop van **multipliciteit** q . Dit wordt gedaan om meer flexibiliteit toe te voegen aan de spline; men kan zo punten creëren waar de klasse van differentieerbaarheid lager ligt. In een knoop van multipliciteit q geldt C^{n-q} -differentieerbaarheid. In het bijzonder is het dus mogelijk om een knik in de kromme te maken, door $q = n$ te kiezen. Let wel op dat de knopen, ondanks dat ze op dezelfde waarde liggen, nog steeds verschillende indices hebben, het zijn dus nog steeds q knopen. Doordat het zowel mogelijk is om de kromme glad te houden, als om knikken te maken, zijn B-splines in de praktijk zo geschikt voor veel toepassingen.

In Figuur 10 zijn enkele basisfuncties gegeven.



FIGUUR 10. BASISFUNCTIES VAN ORDE 1, 2, 3, 4 EN 5 (VAN LINKS NAAR RECHTS).

Van links naar rechts zien we de basisfuncties $B_{0,1}$ tot en met $B_{0,5}$ getekend voor een uniforme knopenverdeling. De basisfunctie $B_{0,j}$ heeft graad $j-1$, oftewel orde j en is C^{j-2} -differentieerbaar. Hierbij is C^{-1} -differentieerbaar gedefinieerd als discontinu. Omdat alle basisfuncties van orde j deze klasse van differentieerbaarheid hebben, is de B-spline van graad n altijd C^{n-1} -differentieerbaar.

De basisfuncties zijn compact gedragen. Preciezer, de drager van de basisfunctie $B_{i,j}$ is gelijk aan het interval $[u_i, u_{i+j}]$. Een gevolg daarvan is dat bij het verplaatsen van een stuurpunt, alleen dát deel van de kromme verandert, waarvoor de basisfuncties actief zijn bij dat stuurpunt en dat betekent de kromme heel gemakkelijk lokaal te veranderen is. Deze eigenschap wordt ook wel **lokale invloed** genoemd. Een enkele Bézierkromme heeft niet deze eigenschap, maar heeft wel de eigenschap dat de verandering van de kromme het grootst is in de buurt van het stuurpunt dat wordt verplaatst. Dit wordt ook wel **pseudo-lokale invloed** genoemd.

Opmerking 6.3.1 De knopen $u_{p+1}, \dots, u_{N-p-1}$ heten de **interne knopen**. Als de beginknoop p -voudig is en de eindknoop ook, en er geen interne knopen zijn, dan is er sprake van een Bézierkromme. [6]

Opmerking 6.3.2 (Terminologie) In ons geval zijn de t_i equidistant verdeeld over het interval $[0, 1]$, oftewel $\forall i \ t_i = i/m$, dan spreken we van een **uniforme B-spline**. Het is ook mogelijk dat de eerste $p+1$ knopen alle nul zijn en de laatste $p+1$ knopen alle één, in dat geval is er sprake van een **aperiodieke B-spline**. Dit wordt meestal gedaan, zodat de kromme begint in het eerste stuurpunt en eindigt in het laatste.

We kunnen het concept van B-splines nog verder veralgemeniseren door aan elk punt een **gewicht** toe te voegen, waardoor de formule voor de B-spline eruit zal zien als

$$\mathbf{x}(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i P_i}{\sum_{i=0}^n N_{i,p}(u) w_i}.$$

*Dit wordt ook wel een rationale B-spline genoemd, voluit een **Niet Uniforme Rationale B-Spline kromme**, dat wordt afgekort tot **NURBS-kromme**.*

7 Splines in de praktijk

In de praktijk is het vaak zo dat een industrieel ontwerper een kromme in gedachten of op tekening heeft, en deze wil benaderen met een B-spline. Ook kan het zo zijn dat een ontwerper een optimale oplossing voor een bepaalde kromme zoekt en een spline gebruikt om die te vinden.

7.1 Eigenschappen van B-splines

B-splines voldoen aan een aantal prettige eigenschappen, die ze geschikt maken voor toepassingen in industrieel ontwerp.

Resultaat 7.1.1 (Eigenschappen van B-splines) *B-splines van graad n voldoen aan de volgende eigenschappen.*

- **Partitie van de eenheid.** *De B-spline basisfuncties van graad n tellen op tot de functie die identiek één is. Daardoor is elk punt op de B-spline-kromme een convexe combinatie van zijn stuurpunten.*
- **Gladheid.** *De kromme is C^{n-1} -differentieerbaar in de knopen. In geval van meervoudige knopen neemt de klasse van differentieerbaarheid in die knopen af. In de praktijk wordt dit gebruikt om knikken te maken. Losse Bézierkrommen hebben deze eigenschap niet.*
- **Lokale invloed.** *Als gevolg van de lokale invloed is het mogelijk om de kromme slechts gedeeltelijk te veranderen door een stuurpunt te verplaatsen. Als een deel van de kromme goed ligt en een ander deel niet, dan kan men het deel dat goed ligt laten liggen door alleen stuurpunten te verplaatsen die geen invloed hebben op het verloop van dat deel van de kromme. Bézierkrommen hebben deze eigenschap niet.*
- **Sterke convexe omhullingseigenschap.** *Een Bézierkromme ligt geheel binnen de convexe omhulling van zijn stuurpunten. Voor B-splines geldt deze eigenschap ook, en bovendien een sterkere variant. Voor elk deelinterval u_i, u_{i+1} geldt dat het stukje B-spline bij dit interval bevat is in de convex omhullende van de stuurpunten $[u_{i-n}, u_i]$. Deze eigenschap is een gevolg van het compact gedragen zijn van de basisfuncties en wordt ook wel de sterke convexe omhullingseigenschap genoemd. Dit geeft ontwerpers een beter inzicht op de plaats waar de punten moeten liggen om een gegeven kromme goed met een B-spline te benaderen.*

- ***Affiene invariantie.** Ook B-splines voldoen aan affiene invariantie. Als gevolg hiervan kan men de kromme gemakkelijk op schaal ontwerpen of isometrisch tekenen.*

In het laatste hoofdstukje zeggen we kort iets over toepassingen van B-splines.

7.2 Toepassingen

B-splines worden in de praktijk toegepast op plaatsen waar het nodig is om een kromme te vinden die aan een bepaalde mate van gladheid voldoet. Voor het ontwerpen van de carrosserieën van auto's worden echter **spline-oppervlakken** gebruikt. Spline-oppervlakken zijn, eenvoudig gezegd, splines in twee variabelen. Deze zijn een stuk geavanceerder dan spline-krommen, de wiskunde over deze objecten is dan ook nog lang niet voltooid. Meer informatie hierover is te vinden in [4]. De wiskunde betreffende de spline-krommen is wel al voor een groot deel "af", maar is veel uitgebreider dan hier is besproken.

Een voorbeeld van een toepassing van B-spline-krommen is een weg of spoorweg. Deze moet voldoende glad zijn opdat de voertuigen die eroverheen gaan niet te scherpe bochten moeten nemen of te plotselinge bewegingen moeten maken. Dit is een voorbeeld van een optimalisatieprobleem, omdat het een vraag is hoe de weg of spoorweg het best kan lopen zodat de bochten niet te scherp zijn. Voor het ontwerp van wegen of spoorwegen worden daarom vaak splines gebruikt: eerst worden de stuurpunten en knopen gekozen en daarna gaat een computerprogramma op zoek naar de plaats van deze stuurpunten en knopen die de beste oplossing geeft [7]. Het absolute optimum is meestal geen spline, maar men kan met splines al snel heel goed in de buurt komen. Omdat splines zo eenvoudig te hanteren zijn, worden ze hier zo vaak voor gebruikt.

Hetzelfde geldt voor de baan van een achtbaan: die moet tenslotte ook aan bepaalde gladheidsvoorwaarden voldoen en mag niet te scherpe bochten maken, denk dan bijvoorbeeld aan het binnen de limieten houden van g-krachten. Bij een achtbaan is het ook belangrijk dat de derde afgeleide geen te hoge waarden aanneemt, aangezien de derde afgeleide staat voor de verandering in de kracht, en het is gevaarlijk en onprettig voor de passagiers als krachten te snel worden opgebouwd.

8 Referenties

- [1] Spline (mathematics), Wikipedia: the Free Encyclopedia.
http://en.wikipedia.org/wiki/Spline_%28mathematics%29
- [2] Eric W. Weisstein. “B-Spline” From *MathWorld*—A Wolfram Web Resource.
<http://mathworld.wolfram.com/B-Spline.html>
- [3] George M. Phillips, *Interpolation and Approximation by polynomials*, Springer Verlag, 2003
- [4] C.R. Traas, H.G. Ter Morsche, R.M.J. Van Damme, *Splines en wavelets*, Epsilon Uitgaven, 2001
- [5] Dictaat *Numerieke Wiskunde*, 2002, Mathematisch instituut, Universiteit Utrecht
- [6] *Bézierkrommen en B-splines*, Erik Leppen en Roeland Warringa, juni 2006, Kaleidoscoop, Mathematisch instituut, Universiteit Utrecht
- [7] *Splines Toegepast*, een lezing door C.R. Traas te Universiteit Utrecht, 28 juni 2006