

link om te commenten (en eventueel deftig de vergelijkingen te zien):

https://docs.google.com/document/d/1pgBRmaKV8SfBWcWRAWAlz_nAEDEk6CcmaphKINTuWsY/edit?usp=sharing

TMI samenvatting

Door Jonas Soenen in academiejaar 2015-2016

Gebaseerd op Cursus Toepassingen van de meetkunde in de informatica door Dirk Roose

En enkele stukjes uit:

- Introduction to algorithms door Thamas H. Cormen, Charles E. Leiserson en Ronald L. Rivest
- Computational geometry in C door Joseph O'Rourke

Met dank aan Michael Vanderstuyft, Jorik Jooker, Jonas Baes, Robin, Bram Vandendriessche en Charlotte Dupont voor aanvullingen en verbeteringen :)

De samenvatting herschrijft heel wat delen uit de cursus korter en hopelijk duidelijker. Maar de cursus heeft heel wat verklarende figuren die niet in deze samenvatting zijn opgenomen! Ook zijn niet alle bewijzen of uitdrukkingen hier opgenomen als er iets ontbreekt wordt er wel verwezen naar de cursus.

Samenvatting is zo goed als volledig!

HF1: inleiding

1.1 Situering en doel

curven, oppervlakken → hoe ga ik deze bijhouden in een computer?

- adhv punten:
 - hoge resolutie bijhouden → veel punten → veel ruimte
 - manipulatie veel rekenwerk → ik moet transformatie toepassen op alle opgeslagen punten
- wiskundig model
 - heeft bovenstaande nadelen niet
 - maar ik moet wel wiskundig model kunnen opstellen (moeilijk voor complexe oppervlakken)

Vereisten voor een wiskundig model:

- compacte beschrijving (weinig parameters of coëfficiënten)
- snelle evaluatie
- geschikt voor interactief ontwerp (vorm bepaald door enkele punten en logisch verband tussen positie van deze punten en de vorm)

2 verschillende situaties:

- bestaande curven of oppervlaktes → deze moeten we voldoende dicht benaderen
- nieuwe curven of oppervlakten → ontwerpen met grafisch tekenpakket → we moeten een wiskundig model zoeken dat aan de opgelegde voorwaarden voldoet

verschillende toepassingen (zie eventueel cursus p3-4)

1.2 Basisbegrippen

punten en vectoren

Curven en oppervlakken → verzameling van punten

Bij opstellen denken we in termen van punten niet in termen van coördinaten van punten!

voor het berekenen is een assenstelsel noodzakelijk MAAR het coördinatensysteem zal geen invloed hebben op de eigenschappen van resulterende curve of oppervlak

→ coördinaat-onafhankelijke methode

punt: een positie → vaak relatief t.o.v andere punten

vector: een richting, een zin en een lengte

verschillende bewerkingen:

ik heb standaard geen assenstelsel maar om te rekenen dus zal ik een assenstelsel kiezen.

- 2 punten aftrekken (positievectoren) dan krijg ik een vector van punt 1 naar punt 2.
onafhankelijk van assenstelsel
- optelling van 2 punten :dit geeft een positievector ten opzichte van de oorsprong
afhankelijk van het assenstelsel
- vermenigvuldigen met een scalar
afhankelijk van het assenstelsel
- translatie (of verschuiving) = punt + een vector
onafhankelijk van assenstelsel
- affiene combinaties lineaire combinatie van verschillende punten met som van alle gewichten gelijk aan 1. Dus in formulevorm:

$$\vec{x} = \sum_{i=1}^m \alpha_i \vec{p}_i \quad \text{met } \alpha_1 + \alpha_2 + \dots + \alpha_m = 1$$

onafhankelijk van assenstelsel

bewijs: een affiene combinatie van punten is een punt

We kunnen $\vec{x}$ schrijven als:

$$\vec{x} = \alpha_1 \vec{p}_1 + \sum_{i=2}^m \alpha_i \vec{p}_i = (1 - \sum_{i=2}^m \alpha_i) \vec{p}_1 + \sum_{i=2}^m \alpha_i \vec{p}_i = \vec{p}_1 + \sum_{i=2}^m \alpha_i (\vec{p}_i - \vec{p}_1)$$

En dit is een punt waar verschillende translaties op toegepast worden en het resultaat daarvan is een punt. ■

Dus er zijn 3 bruikbare bewerkingen:

- 2 punten aftrekken $\rightarrow$ vector
- punt + vector (translatie) $\rightarrow$ punt
- affiene combinatie van punten $\rightarrow$ punt

curven en oppervlakken

curve is een verzameling van punten

vaak niet voor te stellen als functie $y=f(x)$ soms bij regelmatige figuren adhv $F(x,y)=0$

$\rightarrow$ werken met een parameter u (parametervoorstelling)

$$x=f_1(u)$$

$$y=f_2(u) \text{ met } u \in [a, b]$$

kan ook in 3D:

$$x=f_1(u)$$

$$y=f_2(u)$$

$$z=f_3(u) \text{ met } u \in [a, b]$$

Hier heb ik een parameterlijn (een waarde voor u tussen a en b) en elk punt op deze lijn wordt afgebeeld op een punt in de ruimte.

Ik heb hetzelfde principe bij oppervlakken (alleen in 3D) alleen heb ik daar 2 parameters nodig:

$$x=f_1(u,v)$$

$$y=f_2(u,v)$$

$$z=f_3(u,v) \text{ met } u \in [a, b] \text{ en } v \in [c, d]$$

Hier heb ik een parametervlak (een waarde voor u tussen a en b en een waarde voor v tussen c en d) en elk punt van dit vlak wordt afgebeeld op een punt in de ruimte.

segmentatie

Ipv de curve volledig te modelleren met 1 model kan ik ook verschillende curves achter elkaar zetten

elk apart model heeft een lokale parameter + ik heb een globale die loopt over alle curves

In elk snijpunt wil ik een bepaalde continuïteit

→ ik heb verschillende segmenten

ik moet dus ook die globale parameter (u) mappen op de lokale (t) dit doe ik vaak als volgt

$$t = \frac{u - u_i}{u_{i+1} - u_i} \text{ voor } u \in [u_i, u_{i+1}]$$

Dus dit zal ervoor zorgen dat voor $u \in [u_i, u_{i+1}]$ t een waarde aanneemt tussen 0 en 1.

Ik kan hetzelfde doen voor oppervlaktes alleen werk ik hier met knooppuntlijnen → deze vormen rechthoeken → elke rechthoek is een segment en heeft dus zijn eigen wiskundig model.

Ik zal dus beide globale parameters moeten mappen op lokale

$$r = \frac{u - u_i}{u_{i+1} - u_i}$$

$$s = \frac{v - v_i}{v_{i+1} - v_i}$$

figuren p 8

Affiene transformaties

Als alle affiene combinaties van een stel punten afgebeeld wordt op dezelfde affiene combinatie van de beelden van die punten (behoudt affiene lineaire combinaties)

Dus formeel:

$$\vec{x} = \sum_{i=1}^m \alpha_i \vec{p}_i$$

als $\alpha_1 + \alpha_2 + \dots + \alpha_m = 1$ (affiene combinatie van punten) en w is een affiene transformatie dan geldt:

$$w(\vec{x}) = \sum_{i=1}^m \alpha_i w(\vec{p}_i)$$

Stelling: Elke affiene transformatie kan geschreven worden in de vorm:

$$w(\vec{x}) = A\vec{x} + \vec{v}$$

→ elke affiene transformatie is gelijk aan een lineaire transformatie gevolgd door een translatie.

Bewijs van de omgekeerde redenering: een transformatie in de vorm van een lineaire transformatie en een translatie voldoet aan de definitie van een affiene transformatie.

1. We vertrekken van een affiene combinatie van punten waar we een affiene transformatie op toepassen:

$$\sum_{i=1}^m \alpha_i w(\vec{p}_i)$$

2. We vullen hier onze algemene uitdrukking voor een affiene transformatie in:

$$\begin{aligned} & \sum_{i=1}^m \alpha_i (A\vec{p}_i + \vec{v}) \\ &= A \sum_{i=1}^m \alpha_i \vec{p}_i + \sum_{i=1}^m \alpha_i \vec{v} \end{aligned}$$

3. Hier is $\vec{v}$ een constante en kan buiten de sommatie de overblijvende α 's zijn de gewichten van een affiene combinatie en sommeren dus tot 1.

$$\begin{aligned} & \sum_{i=1}^m \alpha_i \vec{p}_i \\ \text{Daarnaast heb ik} & \quad \text{en dit is eigenlijk mijn oorspronkelijke punt } \vec{x}! \text{ Dus:} \\ &= A\vec{x} + \vec{v} \end{aligned}$$

4. En dan kunnen we zien dat we uitkomen op de definitie van een affiene transformatie: Een affiene transformatie toepassen op een affiene combinatie van punten kan ik doen door de transformatie toe te passen op alle punten (en de affiene gewichten gewoon te behouden). ■

Stel dat een bepaalde curve volledig vastligt door een verzameling punten en elk punt van die curve is een affiene combinatie van die punten. Dan moeten we niet elk punt op de curve transformeren, maar gewoon de verzameling punten en daarvan dezelfde affiene combinatie nemen.

→ denk aan Bézier of splines!

Convex omhullende en convexe combinatie

Convexe verzameling (denk aan lineaire optimalisatie voor de mensen die da vak gehad hebben)

een verzameling van punten is convex als voor elk lijnstuk dat 2 punten van M verbindt, alle punten van het lijnstuk tot M behoren.

convex omhullende is de kleinste convexe verzameling die alle punten M omvat (genoteerd als $CH(M)$ of $Hull(M)$)

convexe combinatie: een affiene combinatie waar alle gewichten positief zijn.

Stelling: een convexe combinatie van een eindige verzameling punten behoort tot de convex omhullende van die verzameling

Bewijs door inductie:

start m=1 → Ik heb 1 punt een convexe combinatie daarvan nemen heeft als resultaat dat punt, ook de convex omhullende van dat punt is alleen dat punt. Dus klopt altijd.

inductiestap: als de stelling voor m-1 punten geldt, geldt het dan ook voor m punten?

Dus ik weet dat elke convexe combinatie van punten $\vec{p}_1$ tot en met $\vec{p}_{m-1}$ behoort tot $CH(M)$, bewijs nu dat alle convexe combinaties van punten $\vec{p}_1$ tot en met $\vec{p}_m$ behoren tot $CH(M)$.

Aangezien $\vec{p}_m$ deel is van de verzameling valt het ook binnen de $CH(M)$.

$$\vec{x} = \sum_{i=1}^m \alpha_i \vec{p}_i$$

We willen dus aantonen dat elk punt $\vec{x}$ behoort tot $CH(M)$ (met som van alle alpha's tot 1 en elke alpha > 0)

We bekijken $\sum_{i=1}^m \alpha_i = 1$ met alle $\alpha_i \geq 0$

Als we uit deze sommatie α_m weglaten dan zal het resultaat ervan liggen tussen 0 en 1:

$$0 < \sum_{i=1}^{m-1} \alpha_i = S < 1 \quad \text{als } \alpha_m \neq 0 \text{ en } \alpha_m \neq 1$$

Als we nu een punt $\vec{y}$ als volgt opstellen:

$$\vec{y} = \sum_{i=1}^{m-1} d_i \vec{p}_i \quad \text{met } d_i = \frac{\alpha_i}{S}$$

Dan zijn alle $d_i > 0$ want zo hebben we onze α_i gekozen en S is positief.

Daarnaast is $\sum_{i=1}^{m-1} d_i = 1$ want:

$$\frac{\sum_{i=1}^{m-1} \alpha_i}{S} = 1 \Rightarrow \sum_{i=1}^{m-1} \frac{\alpha_i}{S} = \sum_{i=1}^{m-1} d_i = 1$$

$\vec{y}$ is dus een convexe combinatie van $\vec{p}_1$ tot en met $\vec{p}_{m-1}$ en behoort volgens onze inductiehypothese dus tot CH(M).

Nu is $\vec{x} = S\vec{y} + (1-S)\vec{p}_m$.

Dus $\vec{x}$ is een convexe combinatie van $\vec{y}$ en $\vec{p}_m$. Omdat $\vec{y}$ en $\vec{p}_m$ beide punten zijn van de convex omhullende zal per definitie $\vec{x}$ ook behoren tot de convex omhullende. ■

1.3 Interpolerende veelterm curven

Probleemstelling: ik heb $n+1$ punten gegeven $(\vec{p}_0, \vec{p}_1, \vec{p}_2, \dots, \vec{p}_n)$ en ik wil een curve $\vec{x}(u)$ die exact door al deze punten gaat.

formeel:

Ik zoek $\vec{x}(u)$ waarbij $u \in [u_0, u_n]$ met $\vec{x}(u_i) = \vec{p}_i$ met $u_0 < u_1 < u_2 < \dots < u_n$

de vraag is nu hoe kies ik mijn u 's:

$$u_0 = 0$$

$$u_i = u_{i-1} + \text{afstand}(p_{i-1}, p_i)$$

met afstand een bepaalde afstandsfunctie $\rightarrow$ kiezen we deze gelijk aan euclidische afstand dan krijgen we koordlengteparametrisatie

alternatief is uniforme parametrisatie waarbij $u_i = i$

over het algemeen geeft de koordlengte parametrisatie een beter resultaat dan uniforme

Als we aannemen dat $\vec{x}(u)$ een veelterm is dan krijgen we algemeen:

$$\vec{x}(u) = \vec{a}_0 + \vec{a}_1 u + \vec{a}_2 u^2 + \dots + \vec{a}_n u^n$$

en ik weet dat $\vec{x}(u_i) = \vec{p}_i$ moet gelden

Dus als ik dat invul voor alle u_i 's dan krijg ik n vergelijkingen en dus een stelsel

Bij numerieke wiskunde hebben we bewezen dat dit een unieke oplossing heeft. (determinant van vandermonde $\rightarrow$ matrix a is inverteerbaar $\rightarrow$ juist 1 oplossing)

MAAR de uitgekomen gewichten a hebben hier geen meetkundige betekenis dus werken we ten opzichte van de veeltermen van Lagrange.

Deze veeltermen zijn ook een basis voor de vectorruimte van alle veeltermen van graad n dus

kan ik $\vec{x}(u)$ schrijven als:

$$\vec{x}(u) = \vec{p}_0 L_0^n(u) + \vec{p}_1 L_1^n(u) + \dots + \vec{p}_n L_n^n(u)$$

eigenschappen van L_i^n :

- $\sum_{i=0}^n L_i^n(u) = 1 \Rightarrow$ Dus voor elke u waarde is $\vec{x}(u)$ een affiene combinatie van punten
- $L_i^n(u_j) = 1 \Leftrightarrow j = i$ en $L_i^n(u_j) = 0 \Leftrightarrow j \neq i$ (eigenschap van lagrangeveeltermen)

Een punt op deze curve vinden kan met de volgende uitdrukking: (Aitken) *niet in formularium*

$$\vec{p}_i^{[r]} = (1 - \alpha_{i,r})\vec{p}_{i-1}^{[r-1]} + \alpha_{i,r}\vec{p}_i^{[r-1]} \quad \text{met} \quad \alpha_{i,r} = \frac{u - u_{i-r}}{u_i - u_{i-r}}$$

met $\vec{p}_i^{[0]} = \vec{p}_i$

Werkwijze:

1. Schrijf alle $\vec{p}_i$ s onder elkaar (interpolatiepunten)
2. En bereken uit $\vec{p}_0^{[0]}$ en $\vec{p}_1^{[0]} \rightarrow \vec{p}_1^{[1]}$
3. Doe dit voor die gehele eerste kolom $\rightarrow$ dan krijg ik $\vec{p}_1^{[1]}$ tot en met $\vec{p}_n^{[1]}$.
4. Bereken nu vanuit de 2e kolom de 3e kolom
5. Blijf dit herhalen tot er 1 waarde overblijft: $\vec{p}_n^{[n]}$

eigenschappen:

1. de curve zal schommelen tussen de interpolatiepunten $\rightarrow$ meer en meer wanneer de graad stijgt
2. de gewichten hoeven niet positief te zijn $\rightarrow$ de interpolerende veelterm curve behoort niet tot de convex omhullende van opgegeven punten (bij Bézier en splines wel)
3. een kleine wijziging in punt $\vec{p}_i$ zal vaak grote veranderingen veroorzaken

$\rightarrow$ interpolerende veeltermen worden nauwelijks gebruikt

behalve bij zeer lage graad (beperken tot lineair interpolerende curven of oppervlakken (belangrijk voorbeeld p14 $\rightarrow$ te begrijpen! ook besproken in notities)

HF2 Bézier-curven

Interpolerende veelterm heeft heel veel nadelen dus bekijken we als alternatief Bézier-curven

2.1 Definities

Algemeen voorschrift van een Bézier-curve:

$$\vec{x}(t) = \vec{b}_0 B_0^n(t) + \vec{b}_1 B_1^n(t) + \dots + \vec{b}_n B_n^n(t) \quad \text{met } t \in [0, 1]$$

Met

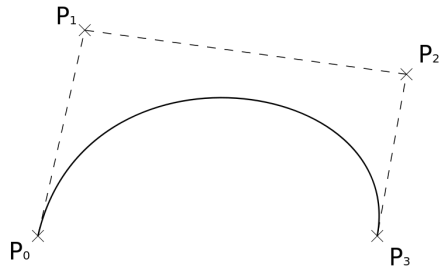
$\vec{b}_i$ = controlepunten of Bézier punten (let op! niet per-se deel van de curve)

$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i \Rightarrow$ de Bernstein veeltermen van graad n

Per definitie

$$B_0^n(t) = 1 \text{ en als } i < 0 \text{ of } i > n \Rightarrow B_i^n(t) = 0$$

open veelhoek gevormd door controlepunten is de controleveelhoek of Bézier-veelhoek



De stippellijn is de controle- of Bézier-veelhoek

2.2 +2.3 Eigenschappen van Bernstein veeltermen en Bézier-curven

Samengenomen omdat het een uit het ander volgt MAAR maak duidelijk onderscheid tussen Bernstein veeltermen en de Bézier curve die ze samen vormen!

$$1. \sum_{i=0}^n B_i^n(t) = 1 \quad \forall t \text{ (sommatie tot 1)}$$

Dus in het algemeen voorschrijf krijg ik een affine combinatie van punten voor elke t
dus het resultaat is een punt!

bewijs adhv binomiaalformule newton (niet in cursus)

$$2. B_i^n(t) \geq 0 \text{ voor } t \in [0, 1] \text{ (positiviteit)}$$

Dit zorgt samen met 1 dat het niet alleen een affine combinatie is van de controlepunten maar ook een convexe combinatie! Dus zal de curve binnen de convex omhullende liggen van de controlepunten.

3. Als $n > 0$:

$$B_i^n(0) = B_i^n(1) = 0 \text{ voor } i \neq 0 \text{ en } i \neq n$$

$$B_0^n(0) = 1; B_i^n(1) = 0 \text{ voor } i \neq 0$$

$$B_i^n(0) = 0; B_i^n(1) = 1 \text{ voor } i = n$$

$\Rightarrow$ De curve interpoleert door het eerste en het laatste punt

dit volgt uit de vorige 2 eigenschappen en het voorschrift van de Bézier-curve

(bij $t=0$ heeft alleen het eerste punt een gewicht van 1 de rest allemaal 0, analoog bij $t=1$)

4. $B_i^n(t) = B_{n-i}^n(1-t)$

bewijs adhv voorschrift Bernsteinveelterm (niet in cursus)

5. $\frac{d}{dt} B_i^n(t) = n * (B_{i-1}^{n-1}(t) - B_i^{n-1}(t))$ (afgeleide van een Bernsteinveelterm)

*bewijs is gewoon afgeleide nemen van voorschrift
(als je kan afleiden tenminste ;)*

6. $\frac{d}{dt} B_i^n(t) = 0 \Leftrightarrow t = \frac{i}{n}$ voor $n > 0$ en $i \in]0, n[$

bewijs is weer gewoon uitrekenen

$\Rightarrow$ de curve zal achtereenvolgens dicht bij $b_0 \dots b_n$ liggen (ik heb een maximum voor $B_i^n(t)$ in $t=i/n$)

7. $B_i^n(t) = (1-t) * B_i^{n-1}(t) + t * B_{i-1}^{n-1}(t)$ (recursiebetrekking)

zie bewijs hieronder

$\Rightarrow B_i^n(t)$ kan geconstrueerd worden via herhaaldelijke lineaire interpolatie
(recursiebetrekking toepassen tot in logisch geval $\rightarrow$ algoritme van Casteljau)

8. curve is van graad n

sommatie van $n+1$ Bernsteinveeltermen (elk van graad n)

9. Verplaatsen van 1 controlepunt zorgt voor een globale verandering van de curve!

$\rightarrow$ De bijhorende Bernstein veelterm is actief over hele interval van $]0;1[$

Bewijs: recursiebetrekking van Bernsteinveeltermen

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

Schrijf bernsteinveeltermen voluit:

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

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

$$= \left[\binom{n-1}{i} + \binom{n-1}{i-1} \right] (1-t)^{n-1-i+1} t^i$$

met tussen de vierkante haakjes:

$$\frac{(n-1)!}{i!(n-1-i)!} + \frac{(n-1)!}{(i-1)!(n-i)!}$$

op gelijke noemer zetten:

$$\begin{aligned} &= \frac{(n-i)(n-1)!}{i!(n-i)!} + \frac{i(n-1)!}{(i)!(n-i)!} \\ &= \frac{(n-i)(n-1)! + i(n-1)!}{(i)!(n-i)!} \\ &= \frac{n!}{(i)!(n-i)!} \\ &= \binom{n}{i} (1-t)^{n-i} t^i = B_i^n(t) \blacksquare \end{aligned}$$

Bézier-curven:

- zachtverlopend
- weinig oscillerend karakter
- variatie verminderingseigenschap: het aantal snijpunten van een willekeurige rechte met de Bézier-curve is kleiner dan of gelijk aan het aantal snijpunten van die rechte met de Bézier veelhoek

2.4 Evaluatie van een punt op een Bézier-curve

gegeven: een Bézier curve $\vec{x}(t)$ en een waarde voor t

gevraagd: punt bijhorende die waarde t op curve $\vec{x}(t)$

→ gebruik algoritme van Casteljau

principe:

1. we vertrekken van een kolom met al onze controlepunten $\vec{b}_0 \rightarrow \vec{b}_n$
2. ik ga hiernaast een kolom maken $\vec{b}_1^{[1]} \rightarrow \vec{b}_n^{[1]}$
3. ik vind $\vec{b}_i^{[1]}$ als $(1-t)\vec{b}_{i-1}^{[0]} + t\vec{b}_i^{[0]}$
4. in het algemeen heb ik $\vec{b}_i^{[k]} = (1-t)\vec{b}_{i-1}^{[k-1]} + t\vec{b}_i^{[k-1]}$
5. dus ik zal kolom per kolom (van links naar rechts) de driehoekige tabel opstellen
6. Totdat ik uiteindelijk bij $\vec{b}_n^{[n]}$ kom en dit is mijn gezochte functiewaarde.
(zelfde principe als gedeelde differenties bij numerieke wiskunde)

Bewijs: Algoritme van Casteljau voor evaluatie van Bézier-curve

We beginnen met algemene uitdrukking voor een Bézier-curve:

$$\sum_{i=0}^n \vec{b}_i B_i^n(t)$$

En hier passen we de recursiebetrekking op toe:

$$\sum_{i=0}^n \vec{b}_i (1-t) B_i^{n-1}(t) + \sum_{i=0}^n \vec{b}_i t B_{i-1}^{n-1}(t)$$

Hierin zijn $B_n^{n-1}(t)$ en $B_{-1}^{n-1}(t)=0$ dus kunnen we de grenzen van onze sommaties aanpassen:

$$\sum_{i=0}^{n-1} \vec{b}_i (1-t) B_i^{n-1}(t) + \sum_{i=1}^n \vec{b}_i t B_{i-1}^{n-1}(t)$$

Nu veranderen we de grenzen van de eerste sommatie van 1 tot n en veranderen we i door i-1

$$\sum_{i=1}^n \vec{b}_{i-1} (1-t) B_{i-1}^{n-1}(t) + \sum_{i=1}^n \vec{b}_i t B_{i-1}^{n-1}(t)$$

En nu kunnen we beide sommaties terug samenvoegen

$$\sum_{i=1}^n \left(\vec{b}_{i-1} (1-t) + \vec{b}_i t \right) B_{i-1}^{n-1}(t)$$

De uitdrukking tussen de haakjes is precies $\vec{b}_i^{[1]}$ bepaalt in het algoritme van Casteljau.

$$\sum_{i=1}^n \vec{b}_i^{[1]} B_{i-1}^{n-1}(t)$$

Nu zouden we bovenstaand proces nog eens kunnen toepassen dan krijgen we:

$$\sum_{i=2}^n \left(\vec{b}_{i-1}^{[1]} (1-t) + \vec{b}_i^{[1]} t \right) B_{i-2}^{n-2}(t) = \sum_{i=2}^n \vec{b}_i^{[2]} B_{i-2}^{n-2}(t)$$

Als we dit n keer doen krijgen we dus:

$$\sum_{i=n}^n \vec{b}_i^{[n]} B_{i-n}^{n-n}(t) = B_0^0(t) \vec{b}_n^{[n]} = \vec{b}_n^{[n]} \blacksquare$$

alternatief:

Stel dat we $\vec{x}(t)$ moeten evalueren voor een groot aantal punten dan kan het efficiënter zijn om de Bézier-curve te herschrijven naar de klassieke veeltermbasis (vervang elke bernstein veelterm door zijn voorschrift en werk uit):

$$\vec{x}(t) = \vec{a}_0 + \vec{a}_1 t + \vec{a}_2 t^2 + \dots + \vec{a}_n t^n$$

En deze dan te evalueren met het hornschema

Hoe bereken ik deze $\vec{a}_i$? **Staat heel onduidelijk in de cursus**
algemene formule:

$$\vec{a}_i = \binom{n}{i} \Delta^i \vec{b}_0$$

met $\Delta \vec{b}_0 = \vec{b}_1 - \vec{b}_0$ en $\Delta^i \vec{b}_0 = \Delta(\Delta^{i-1} \vec{b}_0)$

en eigenlijk is $\Delta \vec{b}_k = \vec{b}_{k+1} - \vec{b}_k$

Dus bijvoorbeeld uitwerking voor $\Delta^2 \vec{b}_0$:

- $\Delta(\Delta \vec{b}_0)$
- $\Delta(\vec{b}_1 - \vec{b}_0)$
- $\vec{b}_2 - \vec{b}_1 - (\vec{b}_1 - \vec{b}_0)$
- $\vec{b}_2 - 2\vec{b}_1 + \vec{b}_0$

2.5 Afgeleiden van Bézier-curven

Niet zo interessant! Gaat gewoon over het berekenen van afgeleiden van Beziercurven en deze evalueren in begin en eindpunt. De eerste en tweede afgeleide van beziercurven staan in het formularium en een punt daarin invullen moet lukken e ;)

$$\frac{d}{dt} \vec{x}(t) = n \sum_{i=0}^{n-1} \Delta \vec{b}_i B_i^{n-1}(t) \quad \text{met } \Delta \vec{b}_i = \vec{b}_{i+1} - \vec{b}_i$$

$\Delta \vec{b}_i$ is een vector dus krijg ik een lineaire (eigenlijk convexe) combinatie van vectoren en dit geeft een vector als resultaat.

Stel dat we werken met een globale parameter u die we eerst nog moeten omzetten naar een lokale parameter t dan krijgen we volgende uitdrukking (pas de kettingregel toe):

$$\frac{d}{dt} \vec{x}(t(u)) = \frac{n}{\Delta u_i} \sum_{i=0}^{n-1} \Delta \vec{b}_i B_i^{n-1}(t) \quad \text{met } \Delta \vec{b}_i = \vec{b}_{i+1} - \vec{b}_i \text{ en } \Delta u_i = u_{i+1} - u_i \text{ met } u \text{ waarden}$$

voor globale parameter. (zie eventueel segmentatie)

Als we deze uitdrukkingen evalueren in begin en eindpunt (t=0 en t=1) krijgen we:

$$\frac{d}{dt} \vec{x}(0) = n(\vec{b}_1 - \vec{b}_0)$$

$$\frac{d}{dt} \vec{x}(1) = n(\vec{b}_n - \vec{b}_{n-1})$$

Analoog bij het geval van globale parameter ik zal alleen u_i en u_{i+1} moeten invullen om te evalueren in begin en eindpunt:

$$\frac{d}{dt} \vec{x}(t(u_i)) = \frac{n}{\Delta u_i} (\vec{b}_1 - \vec{b}_0)$$

$$\frac{d}{dt} \vec{x}(t(u_{i+1})) = \frac{n}{\Delta u_i} (\vec{b}_n - \vec{b}_{n-1})$$

Analoog hieraan kan ik hetzelfde berekenen voor de tweede afgeleide:

$$\frac{d^2}{dt^2} \vec{x}(t) = n(n-1) \sum_{i=0}^{n-2} \Delta^2 \vec{b}_i B_i^{n-2}(t)$$

$$\frac{d^2}{dt^2} \vec{x}(t(u)) = \frac{n(n-1)}{(\Delta u_i)^2} \sum_{i=0}^{n-2} \Delta^2 \vec{b}_i B_i^{n-2}(t)$$

En deze evalueren in begin en eindpunt geeft:

$$\frac{d^2}{dt^2} \vec{x}(0) = n(n-1)(\vec{b}_2 - 2\vec{b}_1 + \vec{b}_0)$$

$$\frac{d^2}{dt^2} \vec{x}(1) = n(n-1)(\vec{b}_n - 2\vec{b}_{n-1} + \vec{b}_{n-2})$$

en met globale parameter krijg ik gewoon extra factor $\frac{1}{(\Delta u_i)^2}$

Zoals eerder aangehaald is dit niet zo belangrijk de uitdrukkingen staan in het formularium :)

2.6 Graadverhoging

In de standaardbasis kan ik elke veelterm van graad n ook schrijven als een veelterm van graad n+1 (voeg gewoon $0x^{n+1}$ toe aan het voorschrift). Omdat alle Bernsteinveeltermen ook een basis vormen voor alle veeltermen moet dit ook kunnen.

maar hoe doe ik dat concreet?

stelling 4 (pagina 23) zegt:

$$\sum_{i=0}^n \vec{b}_i B_i^n(t) = \sum_{i=0}^{n+1} \vec{b}_i^* B_i^{n+1}(t)$$

de n+2 nieuwe Bézier-punten kunnen berekend worden uit de n+1 oude adhv

$$\vec{b}_i^* = \alpha_i \vec{b}_{i-1} + (1 - \alpha_i) \vec{b}_i \text{ met } i = 0, 1, \dots, n+1 \text{ en } \alpha_i = \frac{i}{n+1}$$

merk op: $b_0^* = b_0$ en $b_{n+1}^* = b_n$

*bereken dus gewoon alle b^**

Bewijs:

ik vertrek van de algemene uitdrukking voor een Bézier-curve:

$$\vec{x}(t) = \sum_{i=0}^n \vec{b}_i B_i^n(t)$$

Deze kan ik ook schrijven als:

$$\vec{x}(t) = \left[\sum_{i=0}^n \vec{b}_i \binom{n}{i} (1-t)^{n-i} t^i \right] * (t + (1-t))$$

Hier schrijf ik gewoon $B_i^n(t)$ uit en doen we maal $(t + (1-t)) = 1$

Dit product werken we uit:

$$\sum_{i=0}^n \vec{b}_i \binom{n}{i} (1-t)^{n-i} t^{i+1} + \sum_{i=0}^n \vec{b}_i \binom{n}{i} (1-t)^{n-i+1} t^i$$

Nu vervangen we in de eerste sommatie de grenzen van de sommatie van 1 tot n+1 en schrijven we i-1 ipv i

$$\sum_{i=1}^{n+1} \vec{b}_{i-1} \binom{n}{i-1} (1-t)^{n-i+1} t^i + \sum_{i=0}^n \vec{b}_i \binom{n}{i} (1-t)^{n-i+1} t^i$$

De term binnen de sommatie eerste is nul voor i = 0 en de term binnen de tweede sommatie is nul voor i=n+1

$$\text{want } \binom{n}{-1} = \binom{n}{n+1} = 0 \quad (\text{per definitie van binomiaalgetallen})$$

Dus zullen we deze waarden van i toevoegen aan de grenzen van de sommatie (dat mag omdat deze termen gelijk zijn aan 0)

$$\sum_{i=0}^{n+1} \vec{b}_{i-1} \binom{n}{i-1} (1-t)^{n-i+1} t^i + \sum_{i=0}^{n+1} \vec{b}_i \binom{n}{i} (1-t)^{n-i+1} t^i$$

En nu zijn de grenzen van mijn sommaties gelijk en kan ik deze samennemen dus dan hebben we:

$$\vec{x}(t) = \sum_{i=0}^{n+1} \left[\vec{b}_{i-1} \binom{n}{i-1} + \vec{b}_i \binom{n}{i} \right] (1-t)^{n-i+1} t^i$$

Aan de andere kant kunnen we $\vec{x}(t)$ ook schrijven als een Bézier-curve van graad n+1:

$$\vec{x}(t) = \sum_{i=0}^{n+1} \vec{b}_i^* B_i^{n+1}(t) = \vec{x}(t) = \sum_{i=0}^{n+1} \vec{b}_i^* \binom{n+1}{i} (1-t)^{n+1-i} t^i$$

Als we deze 2 aan elkaar gelijk stellen vinden we dat

$$\vec{b}_i^* \binom{n+1}{i} = \vec{b}_{i-1} \binom{n}{i-1} + \vec{b}_i \binom{n}{i} \quad \text{voor } i = 0..n+1$$

$$\vec{b}_i^* = \frac{\vec{b}_{i-1} \binom{n}{i-1} + \vec{b}_i \binom{n}{i}}{\binom{n+1}{i}}$$

$$= \frac{\vec{b}_{i-1} \frac{n!}{(i-1)!(n-i+1)!} + \vec{b}_i \frac{n!}{i!(n-i)!}}{\frac{(n+1)!}{i!(n+1-i)!}}$$

$$= \frac{\vec{b}_{i-1}i + \vec{b}_i(n-i+1)}{(n+1)}$$

$$\vec{b}_i^* = \frac{i}{n+1} \vec{b}_{i-1} + \left(1 - \frac{i}{n+1}\right) \vec{b}_i$$

En dat is juist de uitdrukking uit de te bewijzen stelling ■

enkele opmerkingen:

- herhaaldelijke toepassing van graadverhoging dan krijg ik een functie van hogere graad die meer aanpasbaar is
- omdat de α 's altijd tussen nul en 1 liggen is elke nieuwe b een convexe combinatie van 2 oude b's dit zorgt ervoor dat de nieuwe b op de rechte ligt tussen de 2 oude b's. Ik kan dit zien als de hoekjes die van de veelhoek worden afgesneden (figuur 2.5 p 25)
- bij herhaaldelijke graadverhoging zal de resulterende Bézier veelhoek convergeren naar de curve
- hieruit volgt ook de variatieverminderingseigenschap:
bij elke graadverhoging zal het aantal snijpunten met de random rechte dalen of gelijk blijven
omdat bij veel graadverhoging de Bézier-veelhoek zal convergeren naar de curve
betekent dit dus dat #snijpunten met de curve $\leq$ #snijpunten met Bézier veelhoek

2.7 Opsplitsing van een Bézier-curve

We willen een curve met $n+1$ controlepunten opsplitsen in 2 delen

dus we nemen van een bepaalde Bézier-curve het stuk dat overeenkomt met $t = [0, c]$

dus dan zijn we op zoek naar een curve $\vec{v}(s)$ die voor $t=[0, c]$ samenvalt met $\vec{x}(t)$

die curve zal er als volgt uitzien:

$$\vec{v}(s(t)) = \sum_{i=0}^n \vec{d}_i B_i^n(s) \quad \text{met } s(t) = t/c$$

s is de lokale parameter en t de globale (t is lokaal in de oorspronkelijke curve)

maar wat zijn deze di's?

$\vec{d}_i = \vec{b}_i^{[i]}$ het i-de element op de diagonaal van de tabel opgesteld d.m.v. het algoritme van de Casteljau voor evaluatie van $\vec{x}(c)$.

Bewijs: Opsplitsen van Bézier-curve

Ik heb gegeven een bepaalde curve $\vec{x}(t)$ en ik heb het segment van $\vec{x}$ voor $t \in [0, c]$ $\vec{v}(s(t))$. Omdat $\vec{v}$ en $\vec{x}$ beide deel zijn van dezelfde unieke veeltermcurve van graad n zijn alle afgeleiden naar t van $\vec{v}$ en $\vec{x}$ aan elkaar gelijk in $t=0$.

Dus formeel:

$$\frac{d^r}{dt^r} \vec{v}(s(t)) = \frac{d^r}{dt^r} \vec{x}(t) \text{ met } t=0 \text{ voor } r=0..n$$

We bekijken dit voor elke aparte waarde van r:

- $r=0$: $\vec{v}(s(0)) = \vec{x}(0)$
of met andere woorden $\vec{d}_0$ moet gelijk zijn aan $\vec{b}_0 = \vec{b}_0^{[0]}$

- $r=1$:
$$\frac{d}{dt} \vec{v}(s(0)) = \frac{d}{dt} \vec{x}(0)$$
$$\frac{n}{c} \sum_{i=0}^{n-1} \Delta \vec{d}_i B_i^{n-1}(0) = n \sum_{i=0}^{n-1} \Delta \vec{b}_i B_i^{n-1}(0)$$
$$\frac{n}{c} (\vec{d}_1 - \vec{d}_0) = n (\vec{b}_1 - \vec{b}_0) \text{ met } \vec{d}_0 = \vec{b}_0$$

Dit verder uitwerken geeft:

$$\vec{d}_1 = (1 - c) \vec{b}_0 + c \vec{b}_1 = \vec{b}_1^{[1]}$$

- $r=2$:
$$\frac{d^2}{dt^2} \vec{v}(s(0)) = \frac{d^2}{dt^2} \vec{x}(0)$$
$$\frac{n(n-1)}{c^2} \sum_{i=0}^{n-2} \Delta^2 \vec{d}_i B_i^{n-2}(0) = n(n-1) \sum_{i=0}^{n-2} \Delta^2 \vec{b}_i B_i^{n-2}(0)$$
$$\frac{n(n-1)}{c^2} (\vec{d}_2 - 2\vec{d}_1 + \vec{d}_0) = n(n-1) (\vec{b}_2 - 2\vec{b}_1 + \vec{b}_0)$$

met $\vec{d}_0 = \vec{b}_0$ en $\vec{d}_1 = (1 - c) \vec{b}_0 + c \vec{b}_1$ kunnen we dit uitwerken tot:

$$\vec{d}_2 = (1 - c)^2 \vec{b}_0 + 2(1 - c)c \vec{b}_1 + c^2 \vec{b}_2$$

en dat blijkt mooi gelijk te zijn aan $\vec{b}_2^{[2]}$

- voor $r > 2$ kunnen we analoog nagaan dat $\vec{d}_r = \vec{b}_r^{[r]}$
- vermits $\vec{v}(s(c)) = \vec{d}_n$ en dit punt moet samenvallen met $\vec{x}(c) = \vec{b}_n^{[n]}$ krijgen we $\vec{d}_n = \vec{b}_n^{[n]}$ ■

Concreet zal ik het algoritme van Casteljau toepassen op mijn originele curve om het punt $t=c$ te vinden. Dan kan ik uit de tabel de bovenste diagonaal aflezen dit zijn de controlepunten van mijn eerste deel.

het rechtse deel van de curve $t=[c,1]$ wordt gevormd door de controlepunten op de onderste diagonaal in dezelfde tabel.

bij herhaaldelijke opsplitsing zal de samengevoegde Bézier-veelhoeken convergeren naar de curve $\vec{x}(t)$

→ dit is een snelle manier om Bézier curven te tekenen (vaak met $c=1/2$) men zal dan de controleveelhoek tekenen ipv de curve zelf

2.8 Samengestelde Bézier-curven

Def: Een samengestelde Bézier-curve over een parameter interval $[u_0, u_p]$ is een stuksgewijze veelterm curve, waarbij ieder segment een Bézier-curve is, gedefinieerd over een van de intervallen $[u_0, u_1], [u_1, u_2], \dots, [u_{p-1}, u_p]$ bepaald door knooppunten $u_0 < u_1 < \dots < u_p$

We hebben dus 1 curve $\vec{x}(u)$ waarbij u de globale parameter is die we gaan mappen naar de

lokale parameter in elk segment $[u_i, u_{i+1}]$ met $\frac{(u-u_i)}{(u_{i+1}-u_i)}$

tussen al deze segmenten willen we een zekere continuïteit:

- C0 continuïteit: het eindpunt van de vorige moet samenvallen met het beginpunt van de volgende
- C1 continuïteit: de afgeleide van de vorige in het eindpunt moet hetzelfde zijn als de afgeleide in het beginpunt van de volgende
- C2 continuïteit: zelfde als C1 maar met 2e orde afgeleide

Dus stel dat we dit willen toepassen op 2 Bézier-curven van graad 3:

$$\vec{x}_0(t(u)) = \sum_{i=0}^3 \vec{b}_i B_i^3(t) \quad \text{en} \quad \vec{x}_1(t(u)) = \sum_{i=0}^3 \vec{b}_i^* B_i^3(t)$$

voorwaarden worden dan:

- C0 continuïteit: $\vec{b}_3 = \vec{b}_0^*$
- C1 continuïteit: $\frac{d}{du} \vec{x}_0(t(u_1)) = \frac{n}{\Delta u_0} \Delta \vec{b}_{n-1}$ en $\frac{d}{du} \vec{x}_1(t(u_1)) = \frac{n}{\Delta u_1} \Delta \vec{b}_0^*$

en deze moeten gelijk zijn aan elkaar dus in dit geval:

$$3 \frac{\vec{b}_3 - \vec{b}_2}{\Delta u_0} = 3 \frac{\vec{b}_1^* - \vec{b}_0^*}{\Delta u_1}$$

Dus met andere woorden $\vec{b}_2, \vec{b}_3$ en $\vec{b}_1^*$ moeten colineair zijn. En hun onderlinge afstand

moet voldoen aan een bepaalde verhouding.

opmerking: als ze alleen meer colineair zijn (onderlinge afstand voldoet niet aan de verhouding) dan spreken we van geometrische eerste-orde continuïteit: G1-continuïteit

- C2 continuïteit:

Hier kunnen we analoog de 2e afgeleide invullen voor eindpunt in $\vec{x}_0$ en voor het beginpunt in $\vec{x}_1$ en deze 2 aan elkaar gelijkstellen. Dan krijgen we:

$$6 \frac{\vec{b}_3 - 2\vec{b}_2 + \vec{b}_1}{(\Delta u_0)^2} = 6 \frac{\vec{b}_2^* - 2\vec{b}_1^* + \vec{b}_0^*}{(\Delta u_1)^2}$$

We hadden bij C1 continuïteit een voorwaarde op de ligging van $\vec{b}_2$ tegenover $\vec{b}_1^*$

Nu leggen we nog een extra voorwaarde op de ligging van $\vec{b}_1$ en $\vec{b}_2^*$

Net zoals bij C1 continuïteit kunnen we deze voorwaarde meetkundig interpreteren
bekijk hiervoor cursus pagina 30-31.

2.9 Ontwerp en modellering van complexe Bézier-curven

pagina 31-34 eens doorlezen: toepassingen en besluit.

HF3: Splinecurven

Hier is het enorm belangrijk dat je het verschil in ziet tussen:

- splinecurve en B-splinefunctie
- controlepunten en knooppunten

Opmerking: de samenvatting is hier vaak uitgebreider of langer dan de cursus. Het doel hiervan is vooral om het eenvoudiger voor te stellen en te helpen om het te begrijpen ipv te zeggen dit is zo probeer zelf in te zien waarom ;)

3.1 Definitie

algemeen voorschrift van een splinecurve:

parameterinterval:

$[u_0; u_p]$ met $u_0 \leq u_1 \leq u_2 \leq \dots \leq u_p$

Dit zijn knooppunten (nut hiervan wordt later duidelijk)

$$\vec{s}(u) = \sum_{i=-n}^{p-1} \vec{d}_i N_i^n(u)$$

$\vec{d}_i$ zijn de controlepunten of de “de Boor-punten” (de ervoor is belangrijk!)

$N_i^n(u)$ is de i-de genormaliseerde B-spline-functie van graad n:

- stuksgewijze veeltermen van graad n
- op elk interval $[u_i; u_{i+1}]$ is $N_i^n(u)$ een veelterm van graad n
- maar functievoorschrift kan op elk interval verschillend zijn

2 interpretaties:

- som van stuksgewijze veeltermen ($N_i^n(u)$) met een bepaalde vectorcoëfficiënt $\vec{d_i}$
- gewogen som van punten $\vec{d_i}$ met gewichten $N_i^n(u)$ (later zullen we zien dat dit een convexe combinatie is)

3.2 B-splines

3.2.1 Definitie van B-splines m.b.v. de recursiebetrekking

Als hij het heeft over B-splines dan heeft hij het over een B-spline functie.

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

met $N_i^0(u) = 1$ als $u \in [u_i; u_{i+1})$ en $N_i^0(u) = 0$ als $u \notin [u_i; u_{i+1})$

(helemaal in formularium)

eerste breuk = verhouding locatie u in actieve interval van N_i^{n-1}

tweede breuk = verhouding locatie u in actieve interval van N_{i+1}^{n-1} , waarbij het stuk rechts van u bekeken wordt om de verhouding te bepalen

aan de hand van deze definitie kan ik elke splinefunctie van bepaalde graad construeren (begin bij graad 0 en bouw op naar graad n $\rightarrow$ zie oefenzitting)

3.2.2 Eigenschappen van B-splines

1. **lokaliteit:**

$$N_i^n(u) = 0 \text{ als } u \notin [u_i; u_{i+n+1}]$$

(volgt uit recursiebetrekking)

2. **positiviteit:**

$$N_i^n(u) \geq 0$$

3. **sommatie tot een:**

$$\sum_{i=-\infty}^{+\infty} N_i^n(u) = \sum_{i=-n}^{p-1} N_i^n(u) = 1 \quad \forall u \in [u_0; u_p)$$

(De eerste B-splinefunctie die actief is op interval $[u_0; u_p]$ is $N_{-n}^n(u)$ want deze is volgens eigenschap 1 actief van u_{-n} tot u_1 . De laatste actieve B-splinefunctie binnen dit interval is $N_{p-1}^n(u)$ want deze is volgens eig 1 actief van u_{p-1} tot u_{p+n} . $\Rightarrow$ sommatie loopt van -n tot p-1. Alle andere splines zijn op dit interval van u gelijk aan 0!)

4. continuïteit:

$N_i^n(u)$ is een C^{n-1} -continue functie (indien er geen meervoudige knooppunten voorkomen)

Dus elke B-splinefunctie:

- verschilt van de nulfunctie (is actief) in n+1 knooppuntintervallen: $[u_i; u_{i+n+1}]$ (lokaliteit)
- wordt bepaald door n+2 knooppunten: $u_i, u_{i+1}, u_{i+2}, \dots, u_{i+n+1}$

In een willekeurig punt van interval $[u_o; u_p]$ zijn er hoogstens n+1 B-splines verschillend van nul. (het punt ligt in een interval $[u_i; u_{i+1}]$ de actieve B-splinefuncties zijn hier: $N_{i-n}^n(t)$ tot en met $N_i^n(t)$ dit zijn er inderdaad n+1)

3.2.3 Invloed van de ligging van de knooppunten

Elke B-spline wordt bepaald door de knooppunten $u_i, u_{i+1}, u_{i+2}, \dots, u_{i+n+1}$

adhv recursiebetrekking:

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

met $N_i^0(u) = 1$ als $u \in [u_i; u_{i+1})$ en $N_i^0(u) = 0$ als $u \notin [u_i; u_{i+1})$

belangrijk: zelf beredeneren waarom dit zo is aan de hand van de recursiebetrekking

$N_i^n(u)$

1. Van welke punten hangt graad 0 af?

$N_i^0(u)$ hangt alleen af van u_i en u_{i+1}

2. Van welke punten hangt graad 1 af?

$$N_i^1(u) = \frac{u - u_i}{u_{i+1} - u_i} N_i^0(u) + \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} N_{i+1}^0(u)$$

via $N_i^0(u)$ hangt af van u_i en u_{i+1}

via $N_{i+1}^0(u)$ hangt af van u_{i+1} en u_{i+2}

in voorschrift zijn dit ook de enige u's die gebruikt worden

$\rightarrow$ hangt af van u_i tot en met u_{i+2}

3. Van welke punten hangt graad 2 af?

$$N_i^2(u) = \frac{u - u_i}{u_{i+2} - u_i} N_i^1(u) + \frac{u_{i+3} - u}{u_{i+3} - u_{i+1}} N_{i+1}^1(u)$$

via $N_i^1(u)$ hangt af van u_i tot en met u_{i+2}

via $N_{i+1}^1(u)$ hangt af van $u_i + 1$ tot en met u_{i+3}

→ hangt af van u_i tot en met u_{i+3}

4. Van welke punten hangt graad n af?

Dus in het algemeen zal dit zijn van u_i tot en met u_{i+n+1}

⇒ Heel dit deel staat niet in de cursus maar ik denk wel dat dit handig kan zijn om te begrijpen hoe het eigenlijk ineen zit. (Kan eenvoudig formeel aangetoond worden met inductie maar ik denk dat dit duidelijker is voor de meeste ;))

de vorm van de B-splinefunctie hangt dus af van de ligging van deze knooppunten

2 speciale gevallen:

- equidistante knooppunten: alle B-splines van een bepaalde graad hebben dezelfde vorm
Aan elkaar gelijk op een verschuiving via de u -as (normaal de x -as) na
- k opeenvolgende knooppunten vallen samen: in de recursiebetrekking worden bepaalde noemers = 0 → we zullen de hele term weglaten
(hier later meer over)

3.2.4 Bewijs van enkele eigenschappen van B-splines

cursus p 42-43

3.3 Eigenschappen van B-splinecurven

uit eigenschappen van B-spline-functies leiden we eigenschappen af voor B-splinecurven

1. sommatie tot 1 → ik heb dus een affiene combinatie van punten
 - dus het resultaat van de uitdrukking is een punt
 - spline curven zijn dus ook invariant onder affiene transformaties
2. bij wijziging van 1 de boor punt zal slechts een deel van de splinecurve wijzigen (volgt uit lokaliteit)
 - ik wijzig een bepaalde $\vec{d}_i$
 - dit verandert het gewicht bij $N_i^n(u)$ (als ik de $\vec{d}_i$'s zie als gewichten bij de B-splines)
 - deze B-splinefunctie is alleen verschillend van nul op het interval $[u_i; u_{i+n+1}]$
 - de curve zal dus alleen veranderen in dit interval
 - daarbuiten heeft de B-splinefunctie geen invloed meer en zal de gewijzigde $\vec{d}_i$ dus ook geen invloed hebben

ik heb een lokale afhankelijkheid tegenover globale afhankelijkheid bij Bézier

3. Spline curven liggen binnen de convex omhullende van controlepunten
sterker nog elk punt van een n-de graads splinecurve ligt binnen de convex omhullende van n+1 de boor punten

- neem een random punt op de curve (geen knooppunt)
- dit punt zal binnen een bepaald knooppunteninterval liggen $(u_j; u_{j+1})$
- op dit interval zijn de spline functies $N_{j-n}^n(u)$ tot $N_j^n(u)$ verschillend van nul
- dus de curve op dit interval heeft als voorschrift:

$$\vec{s}(u) = \sum_{i=j-n}^j \vec{d}_i N_i^n(u) \quad \text{voor } u \in [u_j, u_{j+1})$$

- dit is een convexe combinatie van $\vec{d}_{j-n}$ tot en met $\vec{d}_j$
- elk punt binnen het interval $(u_j; u_{j+1})$ zal dus binnen de convex omhullende van n+1 de boor punten liggen ($\vec{d}_{j-n}$ tot en met $\vec{d}_j$)

Wat als dit wel een knooppunt is?

1. Dus dan heb ik een bepaald knooppunt u_j
2. In dit punt zijn B-spline functies van $N_{j-n}^n(u)$ tot $N_{j-1}^n(u)$ verschillend van nul
3. Dus de curve in dit punt heeft als voorschrift:

$$\vec{s}(u) = \sum_{i=j-n}^{j-1} \vec{d}_i N_i^n(u)$$

4. Dit is een convexe combinatie van $\vec{d}_{j-n}$ tot en met $\vec{d}_{j-1}$
5. Dus het punt dat hoort bij knooppunt u_j ligt in de convex omhullende van n de boor punten ($\vec{d}_{j-n}$ tot en met $\vec{d}_{j-1}$)

3.4 Verband tussen knooppunten, controlepunten en de splinecurve

n-de graads B-splinecurve over parameterinterval $[u_0; u_p)$

1. ik heb p+1 knooppunten en p intervallen
2. welke B-splines zijn er hier verschillend van nul?
3. $N_{-n}^n(u)$ tot en met $N_{p-1}^n(u)$
 - a. dus ik heb n+p de boorpunten $(\vec{d}_{-n}, \dots, \vec{d}_{p-1})$ nodig (1 voor elke splinefunctie)
 - b. elke spline functie heeft ook knooppunteninterval $[u_i; u_{i+n+1}]$ nodig
 - c. dus ik heb knooppunten u_{-n} tot u_{n+p} nodig
 - d. dit zijn er $2n+p+1$

dus voor een splinecurve over interval $[u_0; u_p)$ heb ik u_{-n} tot u_{n+p} knooppunten nodig

en daarbij nog $n+p$ de boorpunten ($\vec{d}_{-n}$ tot en met $\vec{d}_{p-1}$)

3.5 Het effect van collineaire of samenvallende controlepunten

pas op hier gaat het over controlepunten! (d's)

1. $n+1$ de boor-punten $\vec{d}_j$ tot en met $\vec{d}_{j+n}$ die collineair zijn (liggen op 1 rechte)
 - uit de convexiteitseigenschap volgt: als $u \in (u_{j+n}; u_{j+n+1})$ dan ligt de curve binnen de convex omhullende van $\vec{d}_j$ tot en met $\vec{d}_{j+n}$
 - De convex omhullende van collineaire punten is een rechte
 $\rightarrow$ voor $u \in (u_{j+n}; u_{j+n+1})$ ligt de curve op deze rechte
 - OPMERKING alleen voor deze u waarden! niet voor lijnstuk van collineaire punten! (dat zijn nog andere u intervallen waar er ook nog andere splines en dus ook de boor punten actief zijn/ een invloed hebben)
2. $n+1$ de boor-punten $\vec{d}_j$ tot en met $\vec{d}_{j+n}$ die samenvallen
 - uit de convexiteitseigenschap volgt: als $u \in (u_{j+n}; u_{j+n+1})$ dan ligt de curve binnen de convex omhullende van $\vec{d}_j$ tot en met $\vec{d}_{j+n}$
 - in dit geval dus 1 punt
 - dus de splinecurve interpoleert hier
3. n samenvallende de boor-punten $\vec{d}_j$ tot en met $\vec{d}_{j+n-1}$ is reeds genoeg voor interpolatie door dit punt:
 - Ik neem dat $\vec{d}_j$ tot en met $\vec{d}_{j+n-1}$ gelijk zijn aan elkaar en dat $\vec{d}_{j+n}$ verschillend is
 - Dus dan ziet de curve op het interval $[u_{j+n}; u_{j+n+1}]$ er als volgt uit:

$$\sum_{i=j}^{j+n} \vec{d}_i N_i^n(u)$$
 - Deze sommatie splitsen we op in een deel met gelijke controlepunten en het andere controlepunt:

$$\sum_{i=j}^{j+n-1} (\vec{d}_i N_i^n(u)) + \vec{d}_{j+n} N_{j+n}^n(u)$$
 - Als ik deze evalueer in u_{j+n} is $N_{j+n}^n(u) = 1$ en is $\sum_{i=j}^{j+n-1} (\vec{d}_i N_i^n(u))$ een convexe combinatie van $\vec{d}_i$'s

- Maar de $\vec{d}_i$'s die overblijven zijn gelijk dus ik kan deze buiten de sommatie brengen:

$$\vec{d}_j \sum_{i=j}^{j+n-1} (N_i^n(u))$$
- en de sommatie die overblijft is gelijk aan 1. (sommatie tot één van B-splinefuncties)
- Dus de curve evalueren in u_{j+n} zal mij controlepunt $\vec{d}_j$ geven dus daardoor interpoleer ik!

opmerking: in de les zei hij: "als je dit kunt beredeneren dan heb je het echt wel door" Dus zie dat ge dit zelf ook kunt beredeneren!

Dit kan ik gebruiken om te zorgen dat de curve interpoleert in begin en eindpunt!

- beginpunt: kies $\vec{d}_{-n}$ tot en met $\vec{d}_{-1}$ samenvallend $\rightarrow$ dan interpoleert de curve door dit punt bij $u = u_0$
- eindpunt: kies $\vec{d}_{p-n}$ tot en met $\vec{d}_{p-1}$ samenvallend $\rightarrow$ dan interpoleert de curve door dit punt bij $u = u_p$

Bij $n+1$ samenvallende de boor punten ontstaat er een 'singulariteit' Dit kan leiden tot een discontinuïteit in de raaklijn aan de splinecurve in dat controlepunt.

3.6 Het effect van samenvallende knooppunten

pas op hier gaat het over knooppunten! (u's)

- meervoudigheid met k zorgt voor een verlies in continuïteit
 - bij een knooppunt met meervoudigheid k is de splinecurve in dit knooppunt slechts $C(n-k)$ continu.
 - $k=n+1 \rightarrow$ ik krijg een discontinue splinecurve
- samenvallende knooppunten worden gebruikt om interpolatie in begin en eindpunt te krijgen
 - dit doen we door begin en eind knooppunt multipliciteit $n+1$ te geven:
We laten dus $u_{-n}, u_{-n+1}, \dots, u_{-1}, u_0$ en $u_p, u_{p+1}, \dots, u_{p+n}$ samenvallen.
 - en ziet de curve er in $[u_0, u_1)$ als volgt uit:

$$\vec{s}(u) = \sum_{i=-n}^0 (\vec{p}_i N_i^n(u))$$

- Dus als ik deze evalueer in u_0 krijg ik:

$$\sum_{i=-n}^0 (\vec{p}_i N_i^n(u_0))$$

- maar omdat u_0 tot en met u_{-n} allemaal gelijk zijn kan ik de B-splinefuncties evengoed evalueren in u_{-n} :

$$\vec{p}_{-n} N_{-n}^n(u_0) + \sum_{i=-n+1}^0 \vec{p}_i N_i^n(u_{-n})$$

- Hierin zijn alle splines binnen de sommatie nul (u_{-n} valt buiten hun actief gebied) en omdat de sommatie van al deze splines voor elke waarde van u in $[u_0, u_1)$ gelijk moet zijn aan 1 dus moet $N_{-n}^n(u_0) = 1$.

Dus alleen $\vec{p}_{-n}$ blijft over.

- BELANGRIJK: Ik ben niet zeker of dit bewijs klopt of zelfs maar ergens op slaat dus als ge iets anders weet vul maar aan :)

3. n samenvallende knooppunten zijn al genoeg om interpolatie te bekomen

- Als de n knooppunten $u_i, u_{i+1}, \dots, u_{i+n-1}$ samenvallen, dan zal de splinecurve interpoleren door $\vec{d}_{i-1}$ in $u = u_i$. Het bewijs volgt hieronder.

$$\vec{s}(u_i) = \sum_{j=i-n-1}^i \left(\vec{d}_j N_j^n(u_i) \right)$$

- Er geldt wegens lokaliteit: $\vec{s}(u_i) = \sum_{j=i-n-1}^i \left(\vec{d}_j N_j^n(u_i) \right)$. We splitsen deze som op in drie delen en krijgen:

$$\vec{s}(u_i) = \sum_{j=i-n-1}^{i-2} \left(\vec{d}_j N_j^n(u_i) \right) + \vec{d}_{i-1} N_{i-1}^n(u_i) + \vec{d}_i N_i^n(u_i)$$

- De eerste term en de derde term vallen nu weg, omdat de B-splines daar gelijk zijn aan nul (zie eigenschap c, stelling 6 op pagina 42). Er geldt dus:

$$\vec{s}(u_i) = \vec{d}_{i-1} N_{i-1}^n(u_i)$$

Wegens de sommatie tot één eigenschap moet nu gelden dat $N_{i-1}^n(u_i) = 1$, zodat inderdaad geldt dat $\vec{s}(u_i) = \vec{d}_{i-1}$, zoals we wilden bewijzen.

3.7 Voorbeelden

nekeer lezen en bijhorende figuren bekijken

3.8 Het algoritme van de Boor voor de evaluatie van splinecurven

We willen dus weten welk punt er hoort bij een bepaalde u :

We gaan kijken in welk u interval de gezochte waarde ligt

dus stel dat dit $u \in [u_l; u_{l+1})$

dan weet ik dat de actieve splines over dit interval $N_{l-n}^n(u)$ tot en met $N_l^n(u)$ zijn.

Het voorschrift van de curve voor dit segment ziet er dus als volgt uit:

$$s(u) = \sum_{i=l-n}^l d_i N_i^n(u)$$

Nu kan ik dit volgens de Boor berekenen via een algoritme analoog aan Casteljau

$$d_i^{[k]} = (1 - \alpha_{k,i}) * d_{i-1}^{[k-1]} + \alpha_{k,i} * d_i^{[k-1]} \text{ met } \alpha_{k,i} = \frac{(u-u_i)}{(u_{i+n+1-k}-u_i)}$$

de tabel opstellen gebeurt analoog als bij Casteljau

OPMERKING: ik moet natuurlijk vertrekken van de de boor punten die staan bij een actieve spline (d_{l-n} tot en met d_l)

bewijs pagina 50 in de cursus + voorbeeld eens doornemen + figuren!

bewijs:

- Ik wil de curve evalueren in $u \in [u_l; u_{l+1})$
- actieve splines over dit interval zijn $N_{l-n}^n(u)$ tot en met $N_l^n(u)$ (de rest is nul). Dus ik krijg voorschrift:

$$\vec{s}(u) = \sum_{i=l-n}^l \vec{d}_i N_i^n(u)$$

- Hier pas ik de recursieuitdrukking voor splines op toe:

$$\begin{aligned} &= \sum_{i=l-n}^l \vec{d}_i \left(\frac{u - u_i}{u_{i+n} - u_i} N_i^{n-1}(u) + \frac{u_{i+n+1} - u}{u_{i+n+1} + u_{i+1}} N_{i+1}^{n-1}(u) \right) \\ &= \sum_{i=l-n}^l \vec{d}_i \frac{u - u_i}{u_{i+n} - u_i} N_i^{n-1}(u) + \sum_{i=l-n}^l \vec{d}_i \frac{u_{i+n+1} - u}{u_{i+n+1} + u_{i+1}} N_{i+1}^{n-1}(u) \end{aligned}$$

- In de eerste sommatie is $N_{l-n}^{n-1}(u) = 0$ ($i=l-n$) want deze spline is actief over $[u_{l-n}; u_{l-n+n-1})$ en dus niet van $[u_l; u_{l+1})$

In de tweede sommatie is $N_{l+1}^{n-1}(u) = 0$ ($i=l$) want deze spline is actief over $[u_{l+1}; u_{l+1+n-1})$ en dus niet van $[u_l; u_{l+1})$

Dus we passen de grenzen van de sommaties aan om deze splines uit te sluiten:

$$= \sum_{i=l-n+1}^l \vec{d}_i \frac{u - u_i}{u_{i+n} - u_i} N_i^{n-1}(u) + \sum_{i=l-n}^{l-1} \vec{d}_i \frac{u_{i+n+1} - u}{u_{i+n+1} + u_{i+1}} N_{i+1}^{n-1}(u)$$

- In de tweede sommatie herschrijven we de grenzen van i (we doen de grenzen $+1$ en vervangen in de indices i overal door $i-1$):

$$= \sum_{i=l-n+1}^l \vec{d}_i \frac{u - u_i}{u_{i+n} - u_i} N_i^{n-1}(u) + \sum_{i=l-n+1}^l \vec{d}_{i-1} \frac{u_{i+n} - u}{u_{i+n} + u_i} N_i^{n-1}(u)$$

- Nu zijn de grenzen van beide sommaties gelijk en kunnen we ze samennemen tot:

$$\begin{aligned}
 &= \sum_{i=l-n+1}^l \vec{d}_i \frac{u - u_i}{u_{i+n} - u_i} N_i^{n-1}(u) + \overrightarrow{d_{i-1}} \frac{u_{i+n} - u}{u_{i+n} + u_i} N_i^{n-1}(u) \\
 &= \sum_{i=l-n+1}^l \left[\vec{d}_i \frac{u - u_i}{u_{i+n} - u_i} + \overrightarrow{d_{i-1}} \frac{u_{i+n} - u}{u_{i+n} + u_i} \right] N_i^{n-1}(u)
 \end{aligned}$$

- En wat er tussen de haakjes staat is juist $\vec{d}_i^{[1]}$ dus:

$$= \sum_{i=l-n+1}^l \vec{d}_i^{[1]} N_i^{n-1}(u)$$

- Als we dit principe n keer herhalen bekomen we:

$$\sum_{i=l}^l \vec{d}_i^{[n]} N_i^0(u) = \vec{d}_l^{[n]} \blacksquare$$

3.9 Toevoegen van knooppunten en controlepunten

⇒ niet zo eenvoudig!

dit is nuttig als:

- men lokale wijzigingen wilt doen: toevoegen van een knooppunt zorgt dat er een een kleiner deel van de curve zal veranderen bij een de boor punt dat in de buurt ligt.
- continuïteitsvereisten ter hoogte van een knooppunt wilt verminderen → meervoudigheid van het knooppunt verhogen

Een knooppunt toevoegen:

Dus het voorschrift van mijn curve ziet er zo uit

$$\vec{s}(u) = \sum_{i=-n}^{p-1} \vec{d}_i N_i^n(u)$$

En ik wil gaan naar iets in de vorm van:

$$\vec{s}(u) = \sum_{i=-n}^{p^*-1} \vec{d}_i^* N_i^{*n}(u)$$

- waarbij $p^* = p + 1$
- met knooppunten u_i^* voor $i = -n, \dots, n + p^*$
- De Boor-punten $\vec{d}_i^*$ met $i = -n, \dots, p^* - 1$

De knooppunten u_i^* :

→ de originele knooppunten aangevuld met het toegevoegde knooppunt

(zo toegevoegd dat het op de juiste plaats staat)

Dus stel dat het nieuwe knooppunt tussen u_ℓ en $u_{\ell+1}$ ligt:

- $u_i^* = u_i$ voor $i = -n, \dots, \ell$
→ deze blijven ongewijzigd
- u_ℓ^* (het nieuwe knooppunt) $\in (u_\ell, u_{\ell+1})$
- $u_{i+1}^* = u_i$ voor $i = \ell + 1, \dots, p + n$
→ deze blijven ongewijzigd maar hun index is verschoven omdat er eentje extra voor is gekomen

Doordat ik dit punt heb toegevoegd veranderen de splines ook!

De B-splinefuncties $N_i^{*n}(u)$

- als $i < \ell - n$ dan hangen de splines $N_i^{*n}(u)$ alleen af van controlepunten u_{-n}^* tot u_ℓ^*
→ deze knooppunten zijn niet gewijzigd dus ook de splinefuncties blijven hetzelfde!
 $\Rightarrow$ voor $i < \ell - n \Rightarrow N_i^{*n}(u) = N_i^n(u)$ en $\vec{d}_i^* = \vec{d}_i$
- als $i > \ell + 1$ dan hangen de splines $N_i^{*n}(u)$ alleen af van controlepunten $u_{\ell+2}^*$ tot u_{n+p}^*
→ deze knooppunten zijn niet gewijzigd maar de indexen zijn wel veranderd!
 $\Rightarrow$ voor $i > \ell + 1 \Rightarrow N_i^{*n}(u) = N_{i-1}^n(u)$ en $\vec{d}_i^* = \vec{d}_{i-1}$
- als $\ell - n \leq i \leq \ell + 1$ dan hangen de splines dus af van het nieuwe knooppunt $u_{\ell+1}^*$
→ ik zal de nieuwe B-splines en de boor punten moeten afleiden uit volgende gelijkheid:

$$\sum_{i=\ell-n}^{\ell} \vec{d}_i N_i^n(u) = \sum_{i=\ell-n}^{\ell+1} \vec{d}_i^* N_i^{*n}(u)$$

En deze zal ik vinden door een stap toe te passen van het de boor-algoritme voor het evalueren in mijn nieuw knooppunt ($u_{\ell+1}^*$)

en dan is:

$$\vec{d}_{\ell-n}^* = \vec{d}_{\ell-n}$$

$$\vec{d}_{\ell+1}^* = \vec{d}_\ell$$

$$\vec{d}_i^* = \vec{d}_i^{[1]} \text{ voor } i = \ell - n + 1, \dots, \ell$$

//TODO eventueel nog een keer praktisch uitschrijven hoeda ik da doe? zonder uitleg waarom?

gevolgen:

1. Om een B-spline te tekenen kan ik dus gewoon meerdere knooppunten toevoegen → meer controlepunten → controleveelhoek zal convergeren naar de curve
Dus meerdere keren tussen elke 2 knooppunten en nieuw knooppunt invoegen
2. Variatie-verminderingseigenschap voor splinecurven (analoog aan Bézier)
door het toevoegen van knooppunten zal het aantal snijpunten met een willekeurige rechte alleen maar dalen of gelijk blijven. Dus als ik dit oneindig keer doe krijg ik de

curve en geldt het dus dat de curve evenveel of minder snijpunten heeft dan de controleveelhoek.

3. Men kan hetzelfde knooppunt r maal toevoegen:

Dan zal ik $n-1$ controlepunten vervangen door $n-1+r$ nieuwe punten

→ deze nieuwe punten kan ik vinden door r stappen te doen van het De Boor algoritme ipv maar 1.

mijn nieuwe knooppunten zijn dus :

$$\overrightarrow{d_{\ell-n+1}^{[1]}}, \overrightarrow{d_{\ell-n+2}^{[2]}}, \dots, \overrightarrow{d_{\ell-n+r-1}^{[r-1]}}, \overrightarrow{d_{\ell-n+r}^{[r]}}, \dots, \overrightarrow{d_{\ell}^{[r]}}, \overrightarrow{d_{\ell}^{[r-1]}}, \dots, \overrightarrow{d_{\ell}^{[1]}}$$

bekijk vooral visueel waar deze getallen staan in de tabel!

4. Door herhaaldelijke toevoeging kan ik een knooppunt multipliciteit n geven → de curve zal dus interpoleren door het bijhorende de boor-punt
→ nuttig voor begin en eindpunt

3.10 Rationale curven

Tot nu toe gebruikten we veeltermen als basisfunctie:

Dus onze parametervoorstelling zag er zo uit:

$$x=f_1(u)$$

$$y=f_2(u)$$

met f_1 en f_2 veeltermfuncties

Deze zijn niet expressief genoeg! We kunnen bijvoorbeeld geen kegelsneden voorstellen!

MAAR als we voor f_1 en f_2 rationale functies gebruiken kunnen we dit wel!

Rationale Bézier-curven

Een rationale Bézier-curve ziet er als volgt uit:

$$\vec{x}(t) = \frac{\sum_{i=0}^n w_i \vec{b}_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \quad \text{met } t \in [0, 1] \text{ met } w_i \text{ normaal } > 0$$

Dit kunnen we zien als een lineaire combinatie van $n+1$ rationale basisfuncties:

$$\vec{x}(t) = \sum_{i=0}^n \vec{b}_i \left(\frac{w_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \right)$$

Positiviteit en sommatie tot een blijven gelden voor deze basisfuncties:

→ nog altijd convexe combinatie van controlepunten $\vec{b}_i$

→ elk punt van de curve ligt binnen de convex omhullende van de controleveelhoek

Nieuwe eigenschappen:

-verhogen van gewicht w_i zal de curve naar controlepunt $\vec{b}_i$ toe bewegen

-bij alle $w_i = 1$ krijg ik een gewone Bézier-curve

Rationale splinecurven: NURBS

NURBS → niet-uniforme rationale B-splines
curve is van de vorm:

$$\vec{x}(u) = \frac{\sum_{i=-n}^{p-1} w_i \vec{d}_i N_i^n(u)}{\sum_{i=-n}^{p-1} w_i N_i^n(u)} \quad \text{met alle } w_i > 0 \text{ en } u \in [a, b]$$

Dit kan ik weer zien als een lineaire combinatie van n+p rationale basisfuncties:

$$\vec{x}(u) = \sum_{i=-n}^{p-1} \vec{d}_i \left(\frac{w_i N_i^n(u)}{\sum_{i=-n}^{p-1} w_i N_i^n(u)} \right)$$

lokaliteit, positiviteit en sommatie tot een blijven gelden:

→ elk punt op de curve is dus een convexe en lokale combinatie van controlepunten $\vec{d}_i$

Als ik alle $w_i = 1$ neem dan heb ik een gewone B-splinecurve.

Bij het verhogen van gewicht w_i zal de curve lokaal naar $\vec{d}_i$ toe bewegen.

HF4: Modelling van oppervlakken

4.1 Interpolerende tensorproduct-oppervlakken

We zullen een oppervlak beschrijven met volgende uitdrukking:

$$\vec{x}(u, v) = \sum_{i=0}^n \sum_{k=0}^m (\vec{p}_{i,k} \phi_i(u) \psi_k(v)) \quad \text{met } u \in [a, b]; v \in [c, d]$$

waarbij $\phi_i(u)$ en $\psi_k(v)$ basisfuncties zijn (denk aan Bernstein-veeltermen, B-splines of rationale functies)

We bouwen deze uitdrukking stapsgewijs op adhv interpolerende veeltermen adhv lagrange.

1. We beginnen met een interpolerende veeltermcurve door de punten $\vec{p}_i$ met $i = 0..n$ (in 3D)

$$\vec{x}(u) = \sum_{i=0}^n (\vec{p}_i L_i^n(u))$$

2. Als we deze nu door de ruimte laten bewegen waarbij de curve ondertussen van vorm veranderd krijgen we een oppervlak. Dit zullen we doen door de interpolatiepunten $\vec{p}_i$ te laten bewegen in de ruimte. Het pad dat deze punten afleggen beschrijven we ook met

een interpolerende veeltermcurve. Dus dan krijg ik voor elke $\vec{p}_i$:

$$\vec{p}_i(v) = \sum_{k=0}^m \vec{p}_{ik} L_k^m(v) \quad \text{met } v \in [v_0, v_m] \text{ en } i=0..n$$

3. Dus als ik aanneem in de eerste uitdrukking dat deze punten $\vec{p}_i$ kunnen veranderen ifv parameter v zal ik noteren:

$$\vec{x}(u, v) = \sum_{i=0}^n (\vec{p}_i(v) L_i^n(u))$$

4. En vullen we hierin onze uitdrukking voor $\vec{p}_i(v)$ dan krijgen we:

$$\vec{x}(u, v) = \sum_{i=0}^n \left(\sum_{k=0}^m \vec{p}_{ik} L_k^m(v) L_i^n(u) \right)$$

Met andere woorden: We beschouwen n verschillende interpolatiepunten die zich verplaatsen in de ruimte volgens het voorschrift van een bepaalde curve. Door al deze interpolatiepunten kunnen we op elk moment een interpolerende veelterm opstellen. En zo krijgen we dus een oppervlak.

Als we in dit voorschrift een bepaalde parameter constant maken dan krijgen we logischerwijs een curve. Stel we stellen v gelijk aan de constante v_c dan krijgen we volgend voorschrift:

$$\vec{x}(u, v_c) = \sum_{i=0}^n \left(\sum_{k=0}^m \vec{p}_{ik} L_k^m(v_c) \right) L_i^n(u)$$

Alles wat hier tussen de haakjes staat staat eigenlijk vast. Het resultaat hiervan is voor elke i een punt. Dit zijn dan de interpolatiepunten die we zullen gebruiken om onze curve te bepalen.

Een analoge redenering kan gemaakt worden door u vast te kiezen. Alleen zullen we daar de sommaties wat anders moeten organiseren om een soortgelijke uitdrukking te krijgen.

Het oppervlak zal dus interpoleren door alle controlepunten $\vec{p}_{ik}$.

4.2 Tensorproduct Bézier-oppervlakken

We kunnen een analoge redenering maken met Bernsteinveeltermen als basisfuncties:

1. We vertrekken van het voorschrift van een Bézier-curve-segment tussen u_0 en u_1 (in 3D):

$$\vec{x}(r(u)) = \sum_{i=0}^n \vec{b}_i B_i^n(r)$$

2. Nu zullen we deze curve door de ruimte laten bewegen door de controlepunten $\vec{b}_i$ te verschuiven volgens het voorschrift van een Bézier-curve (van graad m):

$$\vec{b}_i(s(v)) = \sum_{k=0}^m \vec{b}_{ik} B_k^m(s)$$

3. En als we deze uitdrukking mee invullen in ons voorschrift uit stap 1:

$$\vec{x}(r(u), s(v)) = \sum_{i=0}^n \sum_{k=0}^m \vec{b}_{ik} B_k^m(s) B_i^n(r)$$

Merk op dat we hier ook direct werken met globale parameters die we mappen naar lokale parameters! Dit is dus het voorschrift van een bepaald oppervlaktesegment! We mappen waarden van de globale parameters u en v naar lokale parameters r en s (met r en s $\in [0, 1]$)
 Probeer goed in te zien hoedat dit oppervlak opgebouwt wordt en hoedat ik de volledige uitdrukking kan interpreteren.

We weten dat de Bézier-curven interpoleren in begin en eindpunt dus:

- $\sum_{k=0}^m \vec{p}_{ik} B_k^m(s)$ interpoleert in begin en eindpunt ($\vec{p}_{i0}$ en $\vec{p}_{im}$)
- $\sum_{i=0}^n \vec{p}_{ik} B_i^n(r)$ interpoleert in begin en eindpunt ($\vec{p}_{0k}$ en $\vec{p}_{nk}$)

Het oppervlak interpoleert pas door het punt als beide curven daar interpoleren!

Dus dan krijg ik als interpolatiepunten $\vec{p}_{00}$, $\vec{p}_{n0}$, $\vec{p}_{0m}$ en $\vec{p}_{nm}$ of met andere woorden de 4 hoekpunten van het oppervlak. Dit zijn de enige punten waardoor het oppervlak interpoleert!

4.2.1 Eigenschappen van Bézier-oppervlakken

- Net zoals bij Bézier-curven is het verband tussen een Bézier-oppervlak-segment en zijn Bézierpunten-net affien invariant. Dit wil zeggen dat als ik een affiene transformatie zou moeten doen op dit oppervlak kan ik ipv het gehele oppervlak te transformeren alleen de controlepunten $\vec{p}_{ik}$ transformeren (zie desnoods hf 1)
- Een Bézier-oppervlak-segment ligt binnen de convex omhullende van het Bézierpunten-net.
- Net zoals bij het interpolerende veeltermoppervlak zou ik hier 1 van de 2 parameters u of v als constant kunnen nemen. Het resultaat is dan een Bézier-curve.
- De randen van het Bézier-oppervlak-segment zijn Bézier-curven met als controlepunten de randpunten van het Bézierpunten-net. (Dit is logisch en volgt uit de vorige

eigenschap: neem s of r gelijk aan 0 of 1)

- Evaluatie van een punt op het Bézier-oppervlak:
 - a. Ik zal de binnenste sommatie eerst uitwerken:
Voor elke i (of k) is dit het voorschrift van een Bézier-curve van graad m (of n)
→ ik gebruik het algoritme van casteljau om elk van deze curven te evalueren in s (of r)
 - b. Dan krijg ik n (of m) controlepunten van een Bézier-curve deze curve zal ik dan evalueren in r (of s). (ook adhv het algoritme van casteljau)

Merk op dat ik zelf kan kiezen wat de binnenste sommatie is en welke Bézier-curven ik dus eerst ga evalueren.

complexiteit: Als ik eerst de Bézier-curven van graad m evalueer zal ik dus n+1 keer een Bézier-curve van graad m evalueren. Het evalueren van 1 veelterm van graad m met casteljau vraagt $O(m^2)$ rekenkost. Dus voor n+1 van deze curven is dat $O(nm^2)$.

Daarna moet ik nog 1 Bézier-curve van graad n evalueren dit kost $O(n^2)$.

⇒ in het totaal $O(nm^2)$ rekenkost.

Merk op dat als we kiezen om eerst de Bézier-curven van graad n te evalueren, we een kost krijgen van $O(mn^2)$. Het is dus het voordeligste om de Bézier-curven van lagere graad eerst te evalueren.

- Stel dat ik nu aan graadverhoging of subdivisie (oppervlak verdelen in 2 verschillende oppervlakken) wil doen dan zal ik de procedures die we gezien hebben moeten toepassen op alle curven in de u en of de v richting.

4.2.2 Afgeleiden van aan Bézier-oppervlak

Hier komt het weer op neer dat ge enkele uitdrukkingen krijgt voor de partiële afgeleiden en dat we die evalueren in begin en eindpunt om later te kunnen gebruiken bij samengestelde bezier oppervlakken.

PAS OP: de algemene uitdrukkingen staan niet in het formularium! Die moet ge zelf kunnen afleiden zo te zien (weet wel dat ge de afgeleide van de basisgevallen wel hebt!)

De eerste partiële afgeleide van het oppervlak naar u:

$$\frac{d}{du} \vec{x}(r(u), s(v)) = \frac{n}{\Delta u_0} \sum_{i=0}^{n-1} \sum_{k=0}^m [\vec{b}_{i+1,k} - \vec{b}_{i,k}] B_i^{n-1}(r) B_k^m(s)$$

Dit evalueren in $u = u_0$ geeft:

$$\frac{d}{du} \vec{x}(r(u_0), s(v)) = \frac{n}{\Delta u_0} \sum_{k=0}^m [\vec{b}_{1,k} - \vec{b}_{0,k}] B_k^m(s)$$

Dit is een veelterm in parameter s met vector-coëfficiënten, die voor elke k gelijk zijn aan de verschilvectoren tussen de eerste 2 Bezierpunten in de u-richting. Probeer de vorige zin echt wel te begrijpen!

En als ik deze uitdrukking dan nog eens evalueer in $v = v_0$ dus $s=0$ dan krijg ik:

$$\frac{d}{du} \vec{x}(r(u_0), s(v_0)) = \frac{n}{\Delta u_0} (\vec{b}_{1,0} - \vec{b}_{0,0})$$

Analoog kunnen we afleiden naar v en dan krijgen we:

$$\frac{d}{dv} \vec{x}(r(u), s(v_0)) = \frac{m}{\Delta v_0} \sum_{i=0}^n (\vec{b}_{i,1} - \vec{b}_{i,0}) B_i^n(r)$$

$$\frac{d}{dv} \vec{x}(r(u_0), s(v_0)) = \frac{m}{\Delta v_0} (\vec{b}_{0,1} - \vec{b}_{0,0})$$

Ik zou ook kunnen kiezen om eerst partieel af te leiden naar v om dan partieel af te leiden naar u en daar dan mijn beginpunten in te vullen, dan krijg ik:

$$\frac{d^2}{dudv} \vec{x}(r(u_0), s(v_0)) = \frac{n * m}{\Delta u_0 \Delta v_0} (\vec{b}_{1,1} - \vec{b}_{1,0} - \vec{b}_{0,1} + \vec{b}_{0,0})$$

$$\text{met } \vec{t} = \vec{b}_{1,1} - \vec{b}_{1,0} - \vec{b}_{0,1} + \vec{b}_{0,0}$$

$$= \vec{b}_{1,1} - (\vec{b}_{1,0} + (\vec{b}_{0,1} - \vec{b}_{0,0}))$$

de draaivector

Deze geeft aan hoever het punt $b_{1,1}$ uit het vlak gevormd door de 3 andere punten ligt. (zie figuur p67)

4.3 Samengestelde Bézier-oppervlakken

C0 continuïteit: Om C0 continuïteit te verkrijgen moeten alle randpunten van beide oppervlakken die overeenkomen met elkaar gelijk zijn:

$$\vec{b}_{n,k}^{(1)} = \vec{b}_{0,k}^{(2)}$$

→ kijk goed na dat je de juiste indexen gebruikt voor de zijde waar ze elkaar raken.

C1 continuïteit in de u of v richting: in de u richting vereist dat de eerste afgeleiden in de u richting gelijk zijn (probeer te koppelen aan geval van gewoon beziercurven):

$$\frac{1}{\Delta u_0} (\vec{b}_{n,k}^{(1)} - \vec{b}_{n-1,k}^{(1)}) = \frac{1}{\Delta u_1} (\vec{b}_{1,k}^{(2)} - \vec{b}_{0,k}^{(2)}) \text{ voor } k=0..m$$

→ ook hier zie dat je de juiste indexen gebruikt voor de zijde waar ze elkaar raken!

totale C1 continuïteit: Hier vereist men naast de C1 continuïteit in de rakende randen dat de draaivectoren aan weerszijden van een gemeenschappelijk hoekpunt ook gelijk zijn. Dus de

partiële afgeleide $\frac{d^2}{dudv} \vec{x}(r(u), s(v))$ moeten voor beide vlakken gelijk zijn geëvalueerd in het gemeenschappelijk hoekpunt.

→ bij 4 aaneensluitende segmenten is er een voorwaarde op de 8 controlepunten die rond het gemeenschappelijk punt liggen.

4.4 Spline-oppervlakken

eens door te lezen p67-68 wordt niet zoveel nieuws geïntroduceerd.