1 Bezir curves

Bezir curves are defined by the control points and the basis function, which is called Bernstein polynomial. Bezir curve is given by the following formula: $\sum_i^n B_i^n(t)$ Where the range of $t = [0,1]$ and $B_i^n = \binom{n}{i}(1-t)^{n-i}t^i$ 1: $B_0^0 = 1$ $\forall(t)$. 2: The value of $t < 0$ or $t > n$ implies that the berstein polynomial is zero. 3 $t = 0$ or $t = 1$ the value of berstein polynomial will be qual to 1. 4:If the interval is not between [0,1] for t, it must be rescaled.

2 Properties

The sum of all nth degree bernstein polynomials(basis) passing through any point for any value of $\sum_i^n B_i n(t) = 1$. Proof: A bernstein polynomail of degree n is given by $((1-t)+t)^n$. (It can be seen why the Bezir curves touch the first and the last point. For the value of t = 0 , the first term give 1 while for the value of t = 1 the second term returns 1). Proof by Induction: Basis case: For $n = 0$ the value of the function will $((1-0)+0) ==> 1$ Because it is proved for the value of $r : r < n$ that the sum of all berstein polynomial passing any point on the interval for any value of $t = 1$ then it must also be valid for n. Induction step: Using binomail theorem of Newton: $(a+b)^n = \sum_i^n \binom{(n}{i)} (a)^{n-i}b^i$. plugging the values we get $\sum_i^n \binom{n}{i} (1-t)^{n-i}t^i = \frac{n!}{i!(n-i)!}(1-t)^{n-i}t^i$

3 Degree of bezier curve

Degree of a bezier curve ,given by $n+1$ control points, is n.

3.1 proof

In each basis function the degree of t is $i + (n-i)$.

4 Bezier curve passes through p_0 and P_n

4.1 Proof

By evaluating the bernstein polynomial($B_i^n = \binom{n}{i}(1-t)^{n-i}t^i$), it can be seen that for the value of $t = 0$ and $t = 1$, the bernstein polynomial is equal is to 1. Which is then multiplied by control points, so it returns the position of first and last point.

5 Derivative of bernstein polynomial

- Derivation of Bernstein polynomial

$$\frac{d}{dt}B_i^n(t) = \frac{d}{dt}\binom{n}{i}\cdot(1-t)^{n-i}\cdot t^i$$

$$= \binom{n}{i}\cdot\left[i\,t^{i-1}(1-t)^{n-i}-(n-i)\,t^i(1-t)^{n-i-1}\right] \quad \textit{product rule}$$

$$= \frac{n!}{(i-1)!(n-i)!}t^{i-1}(1-t)^{n-i}-\frac{n!}{i!(n-i-1)!}t^i(1-t)^{n-i-1}$$

$$= n\left(\frac{(n-1)!}{(i-1)!(n-i)!}t^{i-1}(1-t)^{n-i}-\frac{(n-1)!}{i!(n-i-1)!}t^i(1-t)^{n-i-1}\right)$$

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

6

7 properties of Bezier curves

8 Convex Hull Property

8.1 Proof

Proof with the help of stelling 1 and step 1. The Bezier curve lies within the control polygon of curve.From the first property we know that the sum of all bernstein polynomials is equal to 1.(range of t is between 0 and 1) so multiplying any control-point with bernstein polynomials lies either on line or it intersects both ends if value of $t = 0$ or $t = 1$

8.2 step 1: convex combination of the points

Convex combination of the points is affine combination of the points and the sum of all weights is positive.

8.3 Stelling: 1

Convex combination of the finite poins belongs to CH(M). A convex combination equation has property that no matter what value we choose for t(value should be within range) the new point will lie on the point.Example: $(1-t)A + tB$.

9 Variation Diminishing Property:

No straight line intersects a Bzier curve more times than it intersects the curve's control polyline.This property shows that bezier curves follow their control polygon strictly.It also shows that the curve is much smoother that it's control polygon , on which it is defined.

10 Affine invariance

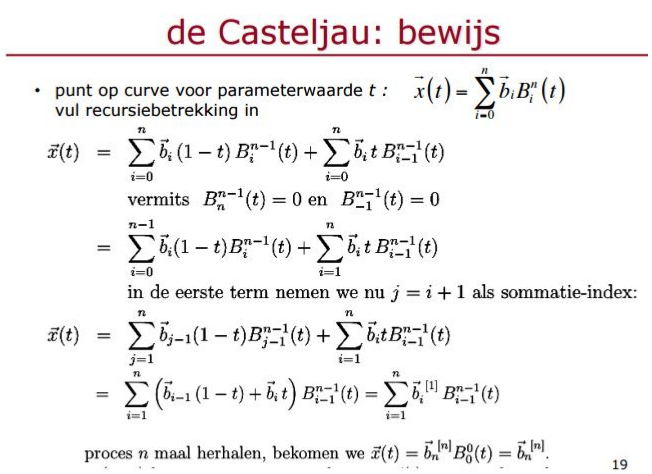
When we want to apply an affine transformation to a bezier curve , we can apply this transformation to the control points of the curve and the result will be same. ==¿ do not need to transform curve.

11 Bezeier curves always interpolate in b_0 and b_n

This happends because for the value of $t = 0andt = 1$ the basis function always returns 1.

12 Finding a point of the bezeier curve

We can find a point on bezier curve with the help of casteljau's algorirthm. Casteljau algorithm works by finding a points on the control polygon lines(points are on affine plan , so we can create convex combination) and then interpolates those points, at some point we find a line which interpolates two points on the line but also intersects the curves at a specific point, this is the point that we are looking for. Algorithm:
say that: $b_i^0 = b_i$ and b_i^k.:
Loop:
k = 1 to n
i = k to n
$b_i^k = (1-t)b_{i-1}^{k-1} + tb_i^{k-1}$


de Casteljau: bewijs

- punt op curve voor parameterwaarde t : $\vec{x}(t) = \sum_{i=0}^{n} \vec{b}_i B_i^n(t)$
  vul recursiebetrekking in

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

vermits $B_n^{n-1}(t) = 0$ en $B_{-1}^{n-1}(t) = 0$

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

in de eerste term nemen we nu $j = i + 1$ als sommatie-index:

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

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

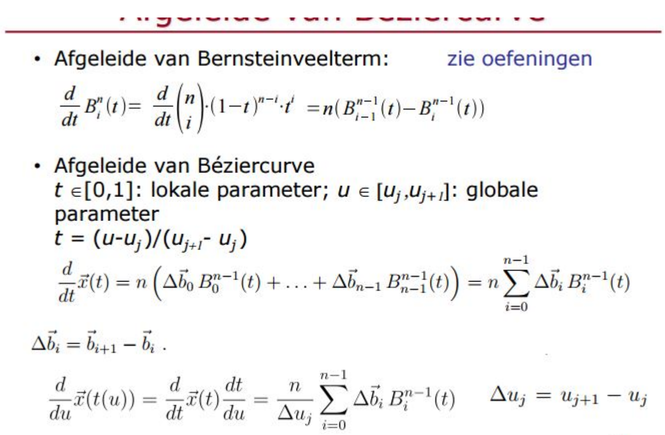
proces n maal herhalen, bekomen we $\vec{x}(t) = \vec{b}_n^{[n]} B_0^0(t) = \vec{b}_n^{[n]}$.

19

$b_n^n = \bar{x}(t)$ Proof:

Special case: For the value of t = 0 in castaeljau's alorithm, the curve will interpolate in first point and this is the point on curve which we were searching. Advantage: Less calculations.

13 Derivative of a Bezier curve



14 Degree evaluation

A bezier curve of degree n is also a bezier curve of degree n+1. We have to introduce a new point on the curves as we increase degree. Increasing the degree degree will result in convergin control polygon to a curve.

15 joining the curve in different segments

15.1 Collinearity:

Two or more than two points are said to be collinear if they lie on same straight line.

15.2 Joining the curves at different segments

When we want to join two curves at some point , we have convert their points from local to global. so an equation for the first segment will be $x_0(t(u)) = \sum_{i=0}^{n} b_i B_i^n(t), u = (u_0, u_p)$ and $x_1(t(u)) = \sum_{i=0}^{n} b_i B_i^n(t), u = (u_p, u_n)$

15.3 Continuety

When we try to join two or more curves at some point , we try to make the curve at that point look smoother.This can be done in number of way.

15.4 C^0 continuety

Say that the last control pont of the first curve is b_3 and first point of second segment is b_0^* then if $b_3 = b_0^*$ then we get C^0 continuety, it is not actually a continuety but it shows that two segments are touching eachother.

15.5 C^1 continuety

Derivative of a curve at some point, gives the slope of a curve on that point.If we place b_0 and b_0^* so that there slope is equal , then we get C^1 contineuty.In other words three points are said to be collinear(b_2, b_0^* and b_1^*).Note: Derivative are taken at the end point of the first curve and at the begining of the second curve.

15.6 C^2 continuety

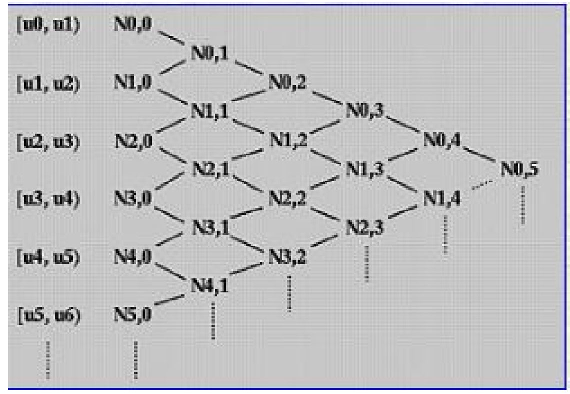
For C^2 continuety ,we have to take twice the derivative of the first segment at the end and second segment at the begining of second segment and those derivative should be equal. For furth explaination , see course.

16 B-Splines

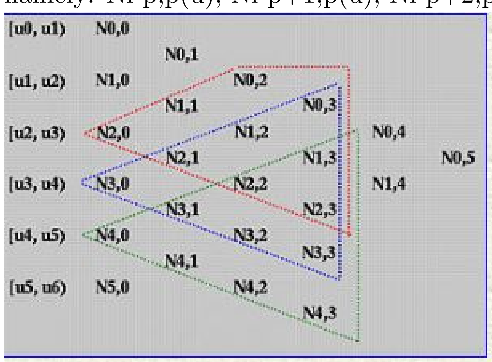
Splines are also combination of Basis function and points.However splines have two interesting properties that they are not non-zero at whole interval , it means that splines are only active at some intervals , for that reason if we move points associated with a specific domain , we will have local effect. The domain is divided in different knots.
The splines are defined as follow:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{aligned}$$

However the domain(U) should be converted to a domain between 0 and 1 and the distance ratio must be same.For a spline of degree n we need $n + 1$ (knots) and n knot-spans.



This can be seen by a triangular schema.
In simple , splines are piecewise continue functions at some domain. We have an important property if we look at the triangular schema:
Basis function Ni,p(u) is non-zero on [ui, ui+p+1). Or, equivalently, Ni,p(u) is non-zero on p+1 knot spans [ui, ui+1), [ui+1, ui+2), ..., [ui+p, ui+p+1).
This property can be seen from triangular schema shown below:
On any knot span [ui, ui+1), at most p+1 degree p basis functions are non-zero, namely: Ni-p,p(u), Ni-p+1,p(u), Ni-p+2,p(u), ..., Ni-1,p(u) and Ni,p(u),



17 Meaning of co-efficients

TODO

18 Properties of Basis function

Locality:
Because a Spline is active at a specific interval, as shown , that splines are not non-zero at whole interval , this shows that they have local control.
On any span [ui, ui+1), at most p+1 degree p basis functions are non-zero, namely: Ni-p,p(u), Ni-p+1,p(u), Ni-p+2,p(u), ..., and Ni,p(u)
Partition of Unity − The sum of all non-zero degree p basis functions on span [ui, ui+1) is 1: The previous property shows that Ni-p,p(u), Ni-p+1,p(u), Ni-

p+2,p(u), ..., and Ni,p(u) are non-zero on [ui, ui+1). This one states that the sum of these p+1 basis functions is 1.For proof , see book:
Each knot of multiplicity k reduces at most k-1 basis functions' non-zero domain.
Consider Ni,p(u) and Ni+1,p(u). The former is non-zero on [ui, ui+p+1) while the latter is non-zero on [ui+1, ui+p+2). If we move ui+p+2 to ui+p+1 so that they become a double knot. Then, Ni,p(u) still has p+1 knot spans on which it is non-zero; but, the number of knot spans on which Ni+1,p(u) is non-zero is reduced by one because the span [ui+p+1,ui+p+2) disappears. This observation can be generalized easily. In fact, ignoring the change of knot span endpoints, to create a knot of multiplicity k, k-1 basis functions will be affected. One of them loses one knot span, a second of them loses two, a third of them loses three and so on.
At each internal knot of multiplicity k, the number of non-zero basis functions is at most p - k + 1, where p is the degree of the basis functions.
Since moving ui-1 to ui will pull a basis function whose non-zero ends at ui-1 to end at ui, this reduces the number of non-zero basis function at ui by one. More precisely, increasing ui's multiplicity by one will reduce the number of non-zero basis functions by one. Since there are at most p+1 basis functions can be non-zero at ui, the number of non-zero basis functions at a knot of multiplicity k is at most (p + 1) - k = p - k + 1. In the above figures, since the multiplicity of knot u6 are 1 (simple), 2, 3, 4 and 5, the numbers of non-zero basis function at u6 are 5, 4, 3, 2 and 1.

19 Coinciding points on curve

TODO

20 Part II

20.1 Algorithm voor discrete meetkundige problemen

In this part,we describe algorithms for discrete collection of objects.More attention is paid to the complexity of these algorithms because for humans a problem can be simple by visualization and however computer has to calculate to take some decisions.

Discrete Giometric Prolems

We divide the problems into two parts , the first part is decision problem. Where the answer is either yes or no. The Second sub-problem is Construction problem or Calculation problem.

20.2 Definitions

We describe problems in two dimensional Euclidean space.Where the distance formula of Euclidean is used.

20.3 Line

A line exist of two points(P_0 and P_1 where the line self is a convex combination of two points.$(1-t)P_0 + P_1$. (A convex combination is a linear combination of points where all points are non-negative and sum to 1.)

20.4 Directed Line

As the name says, a directed line has start point and an end point.

20.5 Velhoek/polygon

In elementary geometry, a polygon /pln/ is a plane figure that is bounded by a finite chain of straight line segments closing in a loop to form a closed polygonal chain or circuit. These segments are called its edges or sides, and the points where two edges meet are the polygon's vertices (singular: vertex) or corners. Notice: If a polygon has N edges , it must have n points.

20.6 Simple Polygon

A polygon is simple if the Polygon side do not overlap(means , they do not have common edges except the end point of the first line and start point of other line).More formally: $\forall(i)$: $P_i\bar{P}_{i+1} \cap P_{i+1}\bar{P}_{i+1} = P_{i+1}$

20.7 Interna and External Polygons

A simple polygon divides the space into two saces , internal(which is nternal to polygon and external which is external to polygon).$Inw(V)$ and $Uitw(V)$.

20.8 Convex set/verzameling

In a Euclidean space (or, more generally in an affine space), a convex set is a region such that, for every pair of points within the region, every point on the straight line segment that joins the pair of points is also within the region.

20.9 Clock-Wise or anti-clockwise

We say that a vector P_0 is clock-wise to a vector P_1 is their vector product is positive because by using vector product/cross product we get a third vector which is perpendicular to both , first and second vector. If the sign is negative , then P_0 is anti-clock wise to P_1.

20.10 Determining whether consecutive segments turn left or right

Determining whether two line at a point p_1 are clock-wise or anti-wise , we can simply calculate the cross-product and it gives the direction of the new segment.same as previous question