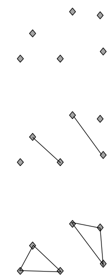
Curves

Modelisation

- Points
  - Defined by 2D or 3D coordinates
- Lines
  - Defined by a set of 2 points
- Polygons
  - Defined by a set of lines
  - Defined by a list of ordered points

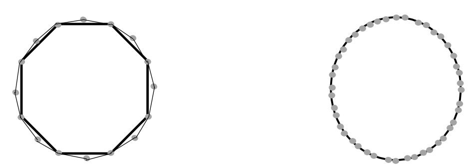


What if you want to have curves?

- Curves are often describe with an analytic equation
- It's different from the discreet description of polygons
- How do you deal with it in Computer Graphics?

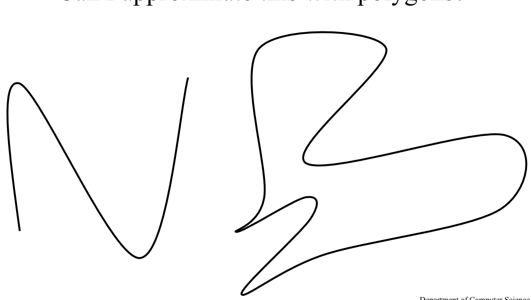
First Solution

- Refine the number of points
  - Can become extremely complex!
  - How do we interpolate?



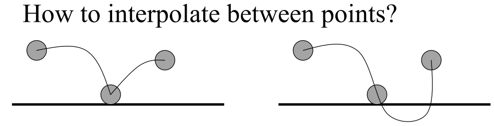
And for more complex curves?

- Can I approximate this with polygons?



Interpolation

How to interpolate between points?



Which one corresponds to what we want?

Math background

- Polynomials
  - Something like

    $a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \ldots + a_n t^n$
- Affine map
  - Something like

    $f(t) = a + bt$

Interpolation

- On affine maps
  - For t in $[t_1,t_2]$

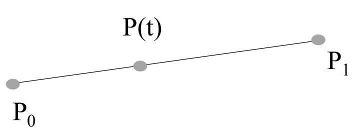
$$t = \frac{t_2 - t}{t_2 - t_1} t_1 + \frac{t - t_1}{t_2 - t_1} t_2$$

$$f(t) = \frac{t_2 - t}{t_2 - t_1} f(t_1) + \frac{t - t_1}{t_2 - t_1} f(t_2)$$

Parameterised line segment

t between [0,1]

$P(t) = P_0 (1\text{-}t) + t P_1$



Multi-Affine maps

- Something like

  $f(t_1,t_2) = c_0 + c_1 t_1 + c_2 t_2 + c_3 t_1 t_2$
- Properties
  - <u>Affine</u> separately on each of its arguments
  - <u>Symmetry</u> when any permutation of the arguments results in the same value

    $f(t_1,t_2) = f(t_2,t_1)$
  - <u>Diagonal</u> when all arguments have the same value

    $f(t,t) = c_0 + A t + B t^2$

Blossoming theorem

- There is a strong connection between multi-affine maps and polynomials
- Every n-argument multi-affine map has a unique nth degree polynomial as its diagonal
- Every nth degree polynomial corresponds to a unique symmetric n-argument multi-affine map, that has this polynomial as its diagonal
- The multi-affine map is called <u>blossom</u> (or <u>polar form</u>)

Interpolation

- Recall interpolation on affine maps

$$f(t) = \frac{t_2 - t}{t_2 - t_1} f(t_1) + \frac{t - t_1}{t_2 - t_1} f(t_2)$$

- Consider $t \in [r,s]$

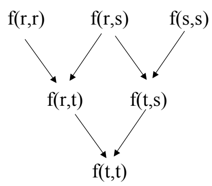
$$f(r,t) = \frac{s - t}{s - r} f(r,r) + \frac{t - r}{s - r} f(r,s)$$

$$f(t,s) = \frac{s - t}{s - r} f(r,s) + \frac{t - r}{s - r} f(s,s)$$

$$f(t,t) = \frac{s - t}{s - r} f(r,t) + \frac{t - r}{s - r} f(s,t)$$

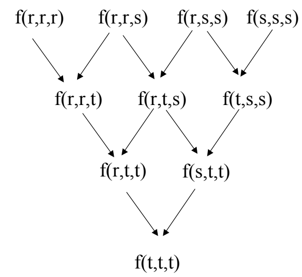
De Casteljau triangles

- Solution for f(t,t)



Can be extend to degree n

- Solution for f(t,t,t)



Bézier curves

f(t1,t2) = (x(t1,t2), y(t1,t2))

Given 3 points:

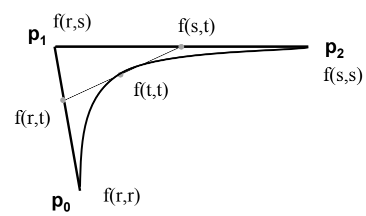
p_0 = f(r,r) p_1 = f(r,s) p_2 = f(s,s)

Interpolation by f(t,t) for any value of t.

All points given by f(t,t) will lie on a curve (2nd degree Bézier curve)

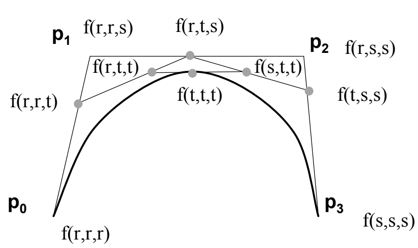
Three control points

- p_0, p_1, p_2



With four control points

- p_0, p_1, p_2, p_3



Properties for f(t,t,t)

- End-points p_0 = P(r) and p_3 = P(S)
- Invariance of shape when change on the parametric interval (affine transformation)
- Convex hull given by the control points
- An affine transformation of the control points is the same as an affine transformation of any points of the curve

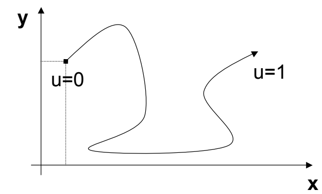
Properties for f(t,t,t)

- If the control points are on a straight line, the Bezier curve is a straight line
- The tangent vector of the curve at end points are (for t=0 and t=1)
  - $P'(r) = 3(p_1 - p_0)$
  - $P'(s) = 3(p_3 - p_2)$

Other way of seeing a Bézier curve

- We drive the position on a curve using a parameter *u* (for 2D or 3D):

$$\boldsymbol{Q}(u) = (\, x(u), y(u) \,)$$



Four control points and [0,1]

- If we restrict t between [0,1]:

$f(t,t,t) = (1-t)^3 f(0,0,0) + 3t(1-t)^2 f(0,0,1) + 3(1-t)t^2 f(0,1,1) + t^3 f(1,1,1)$

- Which can be re-written in a more general case:

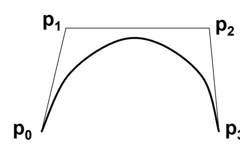
$Q(u) = (\, x(u), y(u) \,) = \Sigma P_i B_i(u)$

Where

$$B_i(u) = \binom{n}{i} u^i (1-u)^{n-i} \qquad \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Bézier Basis

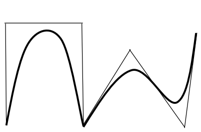
- $b_0(u) = (1-u)^3$
- $b_1(u) = 3\,u\,(1-u)^2$
- $b_2(u) = 3\,u^2\,(1-u)$
- $b_3(u) = u^3$



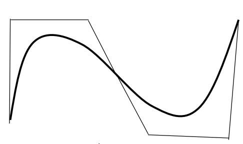
$$\mathbf{Q}(u) = \mathbf{U}\,\mathbf{M_B}\,\mathbf{P} = [u^3\ u^2\ u\ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p_0} \\ \mathbf{p_1} \\ \mathbf{p_2} \\ \mathbf{p_3} \end{bmatrix}$$

Joining Bézier curves

- Better to join curves than raise the number of controls points
  - Avoid numerical instability
  - Local control of the overall shape



C^0



C^1