Outline: Curves

- Piecewise curves
- Hermite curves, Bézier curves and B-splines
- Which to use?
- 3D curves.
- Curved surfaces.

Curves

- Most of the curves used in computer graphics are parametric curves – that is they are based on a certain equation.
- The explicit form of the functions, such as $y = f(x)$, are generally not appropriate because:
  - it is impossible to get multiple y values for a given x value,
  - the form is not rotationally invariant and,
  - you cannot describe curves with a vertical tangent.

Curves

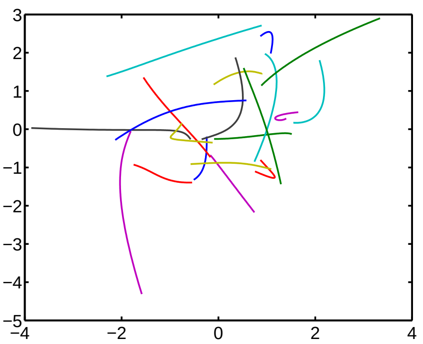
- The implicit form of a function, such as $f(x, y) = 0$, is not very suitable for representing curves either because:
  - the given equation may have more solutions than we want,
  - to restrict the solution to one branch we need extra constraints,
  - joining curves can be a problem.
- The solution is to use parametric functions.

Parametric Curves

- Let $x = x(t)$ and $y = y(t)$ where t is some index – the parameter.
- Piecewise cubic polynomial curve is the most commonly used.
- Individual elements are now cubic functions of t.
- The general form of a curve segment is given by $q(t) = (x(t), y(t))'$ where:

$$
\begin{aligned}
x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \,, \\
y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \,.
\end{aligned}
$$

Parametric Curves



Random cubic parametric curves (coefficients a, b, c, d are random).

Parametric Curves

- Using matrix and vector notation we can write:

$$
\boldsymbol{t} = \begin{bmatrix} t^3 & t^3 \\ t^2 & t^2 \\ t & t \\ 1 & 1 \end{bmatrix} , \; C = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \end{bmatrix} .
$$

  and now $q(t) = C\boldsymbol{t}$.
- To join segments we ensure continuity and smoothness by matching the tangents or derivatives of the curves at the joining points.

Parametric Curves

- To join segments we ensure continuity by computing:

$$
\frac{\partial q(t)}{\partial t} = \left(\frac{\partial x(t)}{\partial t}, \frac{\partial y(t)}{\partial t} \right) = \frac{\partial (C\boldsymbol{t})}{\partial t} = C \frac{\partial \boldsymbol{t}}{\partial t} \,,
$$

  where:

$$
\frac{\partial \boldsymbol{t}}{\partial t} = \begin{bmatrix} 3t^2 & 3t^2 \\ 2t & 2t \\ 1 & 1 \\ 0 & 0 \end{bmatrix} .
$$

- If the last point of curve 1 is the first point of curve 2 and the derivatives are equal at this point they will join smoothly.

Continuity

- It is possible to define many types of continuity:
  - G^0 geometric continuity - the curves join
  - G^1 geometric continuity - the curves join with equal tangent directions.
  - C^1 continuity - the curves join with equal tangent directions and magnitude (first derivatives equal).
  - C^n continuity - the curves join with equal n'th derivatives.
- Which is desired will depend on the context.
- We are basically solving a system of (linear) equations when we compute the coefficients from the given constraints.

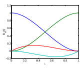
Hermite curves

- A Hermite polynomial form is specified by the definition of two end points $\boldsymbol{p}_1$, $\boldsymbol{p}_4$ and two end tangent vectors, $\boldsymbol{r}_1$, $\boldsymbol{r}_4$.
- Expand C into two matrices:

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x = C_x \boldsymbol{t} = G_x M_H \boldsymbol{t} \ ,$$

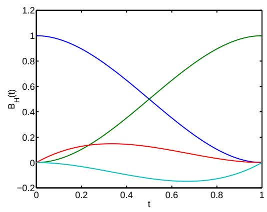
- G_x is the x-component of the geometry matrix.
- M_H is the Hermite basis matrix.
- G_x is $\left[p_{1,x} p_{4,x} r_{1,x} r_{4,x}\right]$.
- Thus we can derive M_H (not shown).

Hermite curves



- We can now write $x(t)$ in two ways: $a_x t^3 + b_x t^2 + c_x t + d_x$, or $G_x B_H(t)$ where $B_H(t) = M_H \boldsymbol{t}$ are the Hermite blending functions.
- Now given the geometry matrix, G, and M_H, we can compute $C = GM_H$.
- To draw the curves we simply evaluate $x(t)$ and $y(t)$.

Hermite curves



- The Hermite basis matrix is:

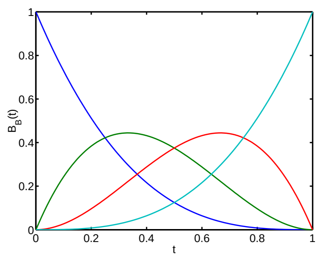
$$M_H = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} .$$

Bézier curves

- Bézier curves are very similar to Hermite curves except that we use four control points, $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$ and $\boldsymbol{p}_4$.
- $\boldsymbol{r}_1 = 3(\boldsymbol{p}_2 - \boldsymbol{p}_1)$ and $\boldsymbol{r}_4 = 3(\boldsymbol{p}_4 - \boldsymbol{p}_3)$.
- G_x is $\left[p_{1,x} p_{2,x} p_{3,x} p_{4,x}\right]$.
- This means that the Bézier basis matrix will be:

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} .$$

Bézier curves



The Bézier blending functions.

Drawing curves

- To draw parametric curves we need to know the so called control points (e.g. $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$ and $\boldsymbol{p}_4$ for a Bézier curve).
- Then we can use $x(t) = G_x M_H \boldsymbol{t}$, $y(t) = G_y M_H \boldsymbol{t}$ to compute the location of the curve for any t value.
- In practice we loop over t (from 0 to 1) in small increments and connect the short line segments.
- If the increment is small enough we can just set the pixel (bad idea?).

Bézier curves

- If we need to join two Bézier curves, with control points $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$, $\boldsymbol{p}_4$ (in both), $\boldsymbol{p}_5$, $\boldsymbol{p}_6$, $\boldsymbol{p}_7$ then we need an extra condition.
- Having $\boldsymbol{p}_4$ in both curves ensures G^0 or C^0 continuity.
- We have:
  - G^1 continuity if $\boldsymbol{p}_3 - \boldsymbol{p}_4 = k(\boldsymbol{p}_4 - \boldsymbol{p}_5)$ where $k > 0$,
  - C^1 continuity if $\boldsymbol{p}_3 - \boldsymbol{p}_4 = \boldsymbol{p}_4 - \boldsymbol{p}_5$.
- This gives us the recipe for constructing arbitrarily long curves with the desired properties.
- The curve is contained within the convex hull of the points.

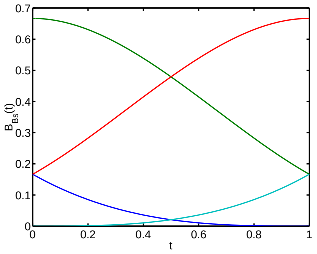
B-splines

- Splines have a long history in computer graphics.
- The natural cubic spline has C^2 continuity and is thus smoother than the Hermite and Bézier curves. The B stands for Basis.
- Geometry matrix: $G_{Bs,j} = \left[\boldsymbol{p}_{j-3} \boldsymbol{p}_{j-2} \boldsymbol{p}_{j-1} \boldsymbol{p}_j\right]$.
- B-spline basis matrix is given by:

$$M_{Bs} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} .$$

- We can derive the basis (blending) functions $B_{Bs}(t)$.

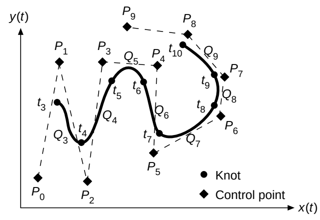
B-splines



The B-spline blending functions.

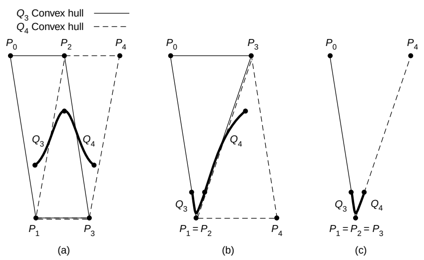
B-splines

- The effect of each (control) point is local.
- B-splines share control points across segments:



B-splines

- We can use duplicated control points to change the continuity of the B-spline:



Other Splines

- Non-uniform splines are more flexible:
  - interpolate, continuity and can easily add points
- Rational splines,

$$x(t) = \frac{X(t)}{W(t)}, \quad y(t) = \frac{Y(t)}{W(t)},$$

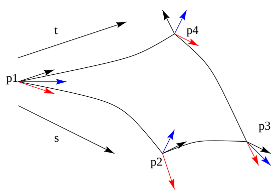
  can be arbitrarily transformed (including the perspective projection).
- Can define all conic sections.
- NURBS (Non-Uniform Rational B-Splines) curves are widely used in computer graphics.

Comparison of curves

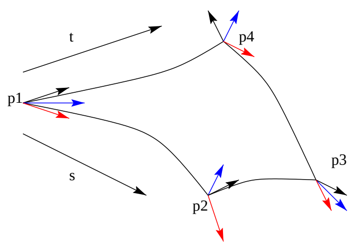
- There are several criteria by which we can select a method including:
  - degree of continuity of the complete curve,
  - speed of computation to generate the curve,
  - ease of definition of the curve,
  - ability of the curve to represent desired objects.
- Distinction not so crucial because we can rewrite the uniform curves.
- Often use different representations in one program.

Curves in 3D and surfaces

- Simple add an extra polynomial $z(t)$ to give us the behaviour in the third dimension.
- Parametric bicubic surfaces use two parameters s and t to parametrise the cubic patch.

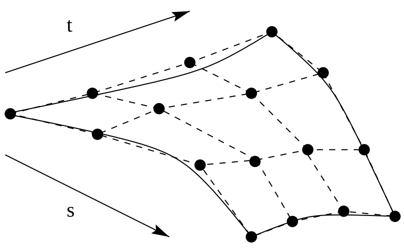


Curves in 3D and surfaces



- Hermite surfaces: define the vertices of the four sided polygon, together with the tangents along the curves in the t and s directions and the twist – 16 conditions to specify a single cubic patch.

Curved surfaces



- Bézier surfaces: defined by 16 control points.

Curved surfaces

- Parametric surface, $q(s,t)$, the tangents to the surface in the s and t directions are:

$$\frac{\partial q(s,t)}{\partial s} \quad \text{and} \quad \frac{\partial q(s,t)}{\partial t},$$

- The normal to the surface is easy to compute and is:

$$\frac{\partial q(s,t)}{\partial s} \times \frac{\partial q(s,t)}{\partial t},$$

- We can display the surface by fixing one of s or t and incrementing the other (in small steps).

Summary

- Having finished this lecture you should:
  - be able to use parametric functions;
  - understand how curved objects are represented in computer graphics;
  - be able to draw a curved object given a series of control points;
  - extend curved objects into 3D and patches.
- OpenGL implements curves but we will not explore this in the labs.