Xi'an Jiaotong-Liverpool University
西交利物浦大学

CPT205 Computer Graphics

Parametric Curves and Surfaces

Lecture 06
2022-23

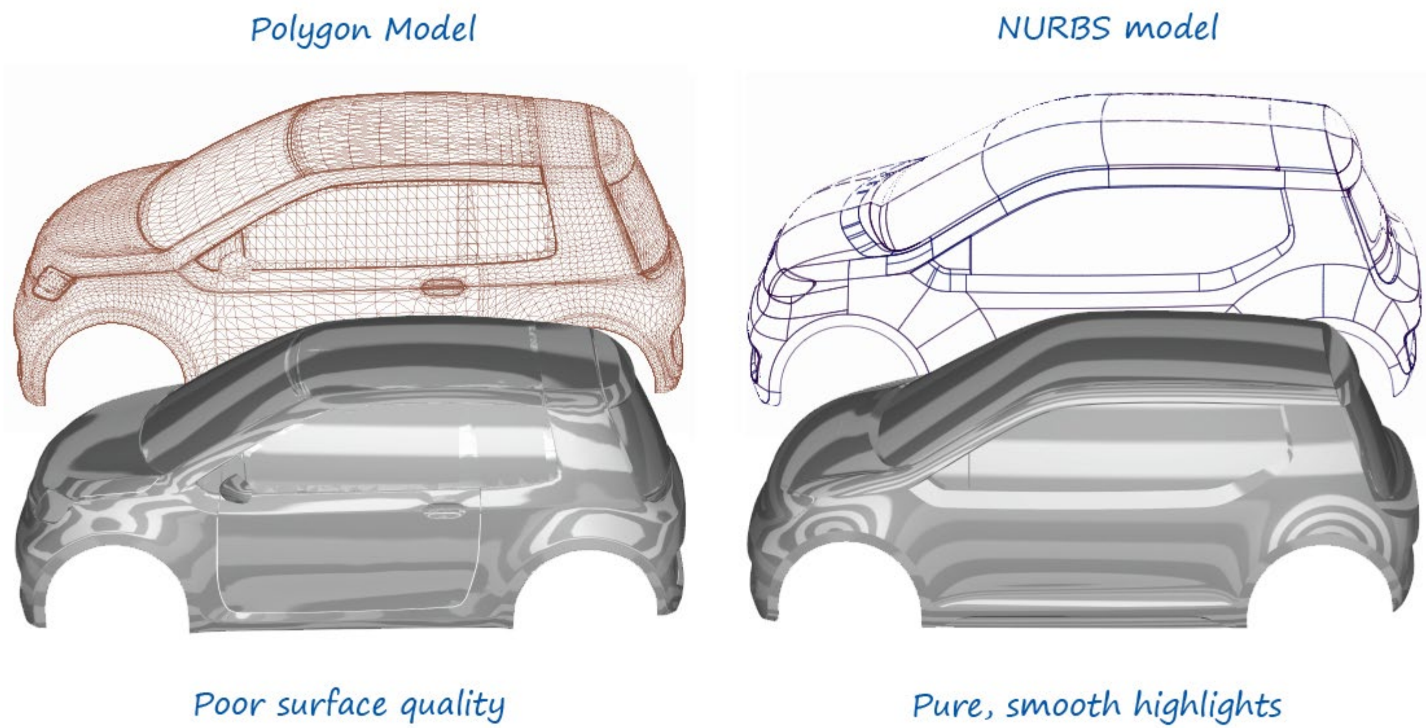
Yong Yue

Topics for today

- Why parametric
- Parametric curves
- Splines
- Revolved, extruded and swept surfaces
- Tensor product surfaces
- Summary

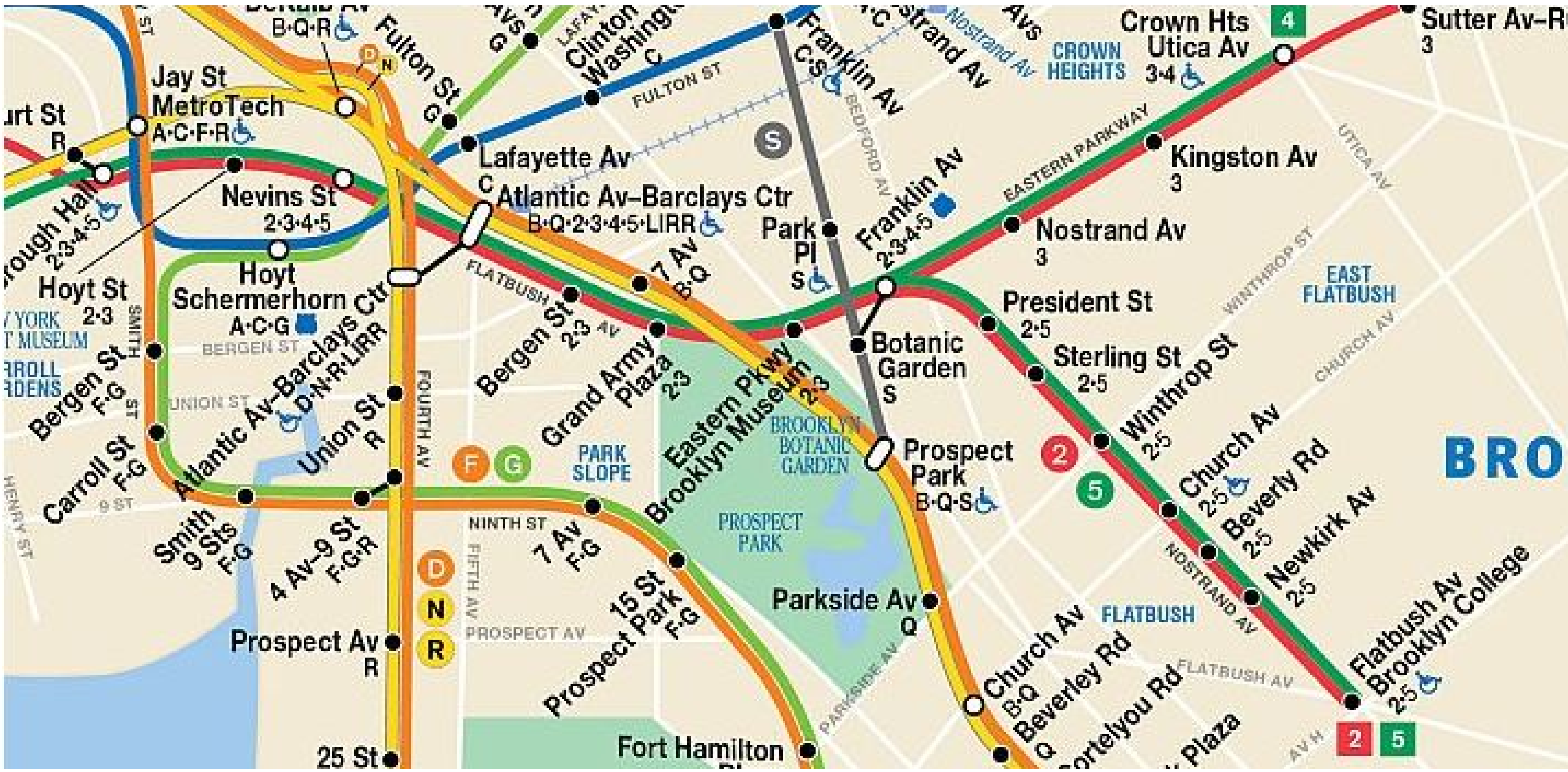
Motivation

- ➢ More realistic 3D rendering of naturally curved objects



Motivation

➢ Easier to follow than poly-lines



Why parametric?

- Parametric surfaces are surfaces that are usually parameterised by two independent variables.
- By parameterisation, it is relatively easy to represent surfaces that are self-intersecting or non-orientable.
- It is impossible to represent many of these surfaces by using implicit functions.
- Even where implicit functions exist for these surfaces, the tessellated representation is often incorrect.

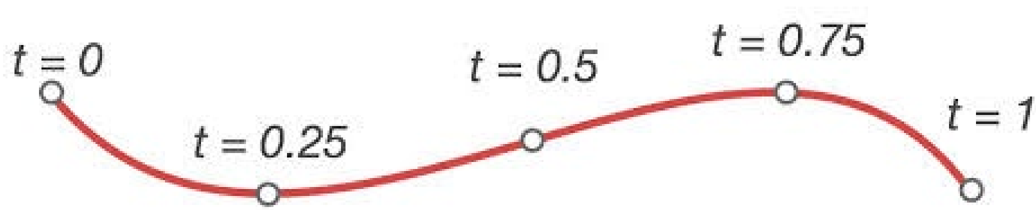
Parametric curves

- A curve in a 2D (x, y) surface is defined as:

  $x = x(t)$
  $y = y(t)$

  where t is a parameter in [0,1].

- In this way the curve is well defined, each value of t in [0,1] defining one and only one point.
- The curve description will not change when rotation occurs.



Parametric equation of a straight line

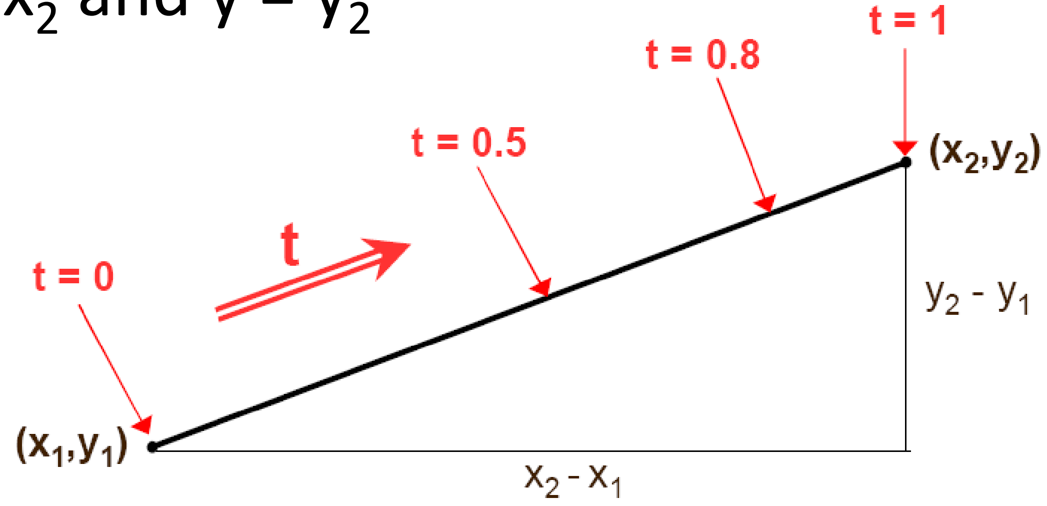
Implicit representation: $y = a_0 + a_1 x$

Parametric (explicit) representation:

$$x = x_1 + t(x_2 - x_1) \qquad (0 \le t \le 1)$$
$$y = y_1 + t(y_2 - y_1)$$

when $t = 0$, $x = x_1$ and $y = y_1$

when $t = 1$, $x = x_2$ and $y = y_2$



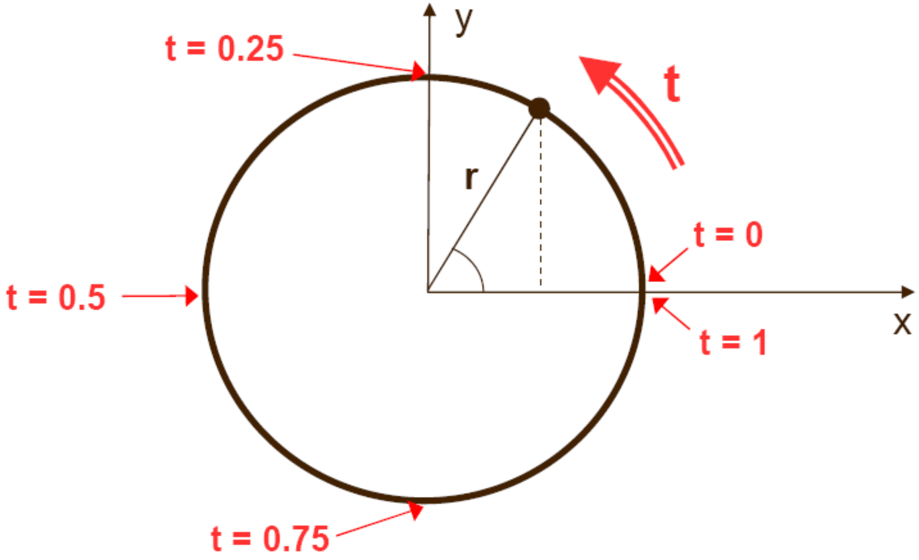
Parametric equation of a circle

Implicit representation:

$x^2 + y^2 = r^2$ (r = radius)

Parametric equation:

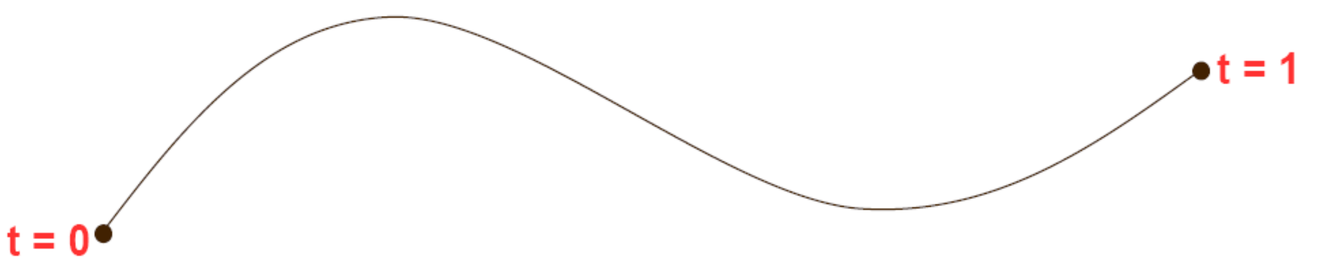
$x = r\cos(360t)$, $y = r\sin(360t)$, $(0 \le t \le 1)$



Parametric equation of a cubic curve

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \qquad (0 \le t \le 1)$$
$$y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3$$

where a_i and b_i terms are constants that vary from curve to curve.



What type of curve to use?

- A curve description should be used, which allows rapid computation (i.e. functions such as sin, cos, exp, log, and so on should be avoided).

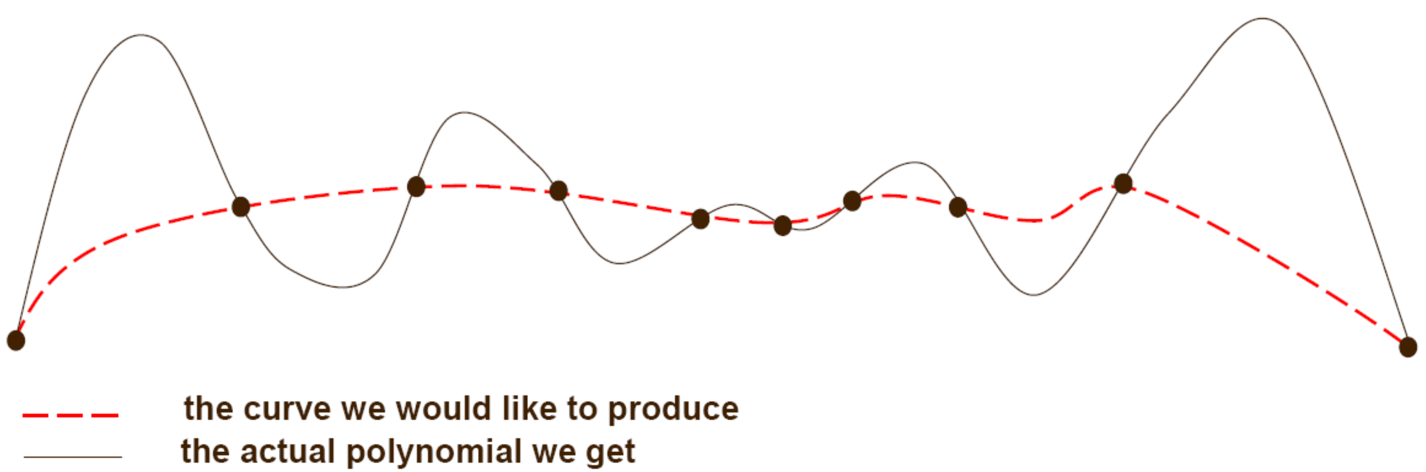
- A polynomial can therefore be used

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, + \ldots + a_n t^n$$

- For interpolation, if there are k points, then n = k – 1 must be chosen, in order to find the correct values for a_i.

Interpolation through k points

- When k = 2, i.e. n = k – 1 = 1; a straight line can be fitted.
- When k = 3, i.e. n = k – 1 = 2; a parabola can be fitted.
- When k is large, n must be large, too; high-degree polynomials (i.e. with a large n) oscillate wildly, particularly near the ends of the line, and are not suitable.



Low-degree polynomial curves

- Polynomials have to be used for efficiency.
- High-degree polynomials are not suitable because of their behaviour.
- For curves of a large number of points
  - they can be broken into small sets
    (e.g. 4 points in each set).
  - a low-degree polynomial is put through each set
    (a cubic for 4 points).
  - these individual curves (cubics) are joined up
    smoothly.
- This is the basis of splines.

Splines

- A spline curve consists of individual components, joined together smoothly, looking like a continuous smooth curve.

- Different types of continuity exist and the following are generally required
  - continuity of the curve (no breaks)
  - continuity of tangent (no sharp corners)
  - continuity of curvature (not essential but avoids some artefact from lighting)

- Each component is a low-degree polynomial, and for these continuities, cubic polynomials are generally needed.

Interpolation and design curves (1)

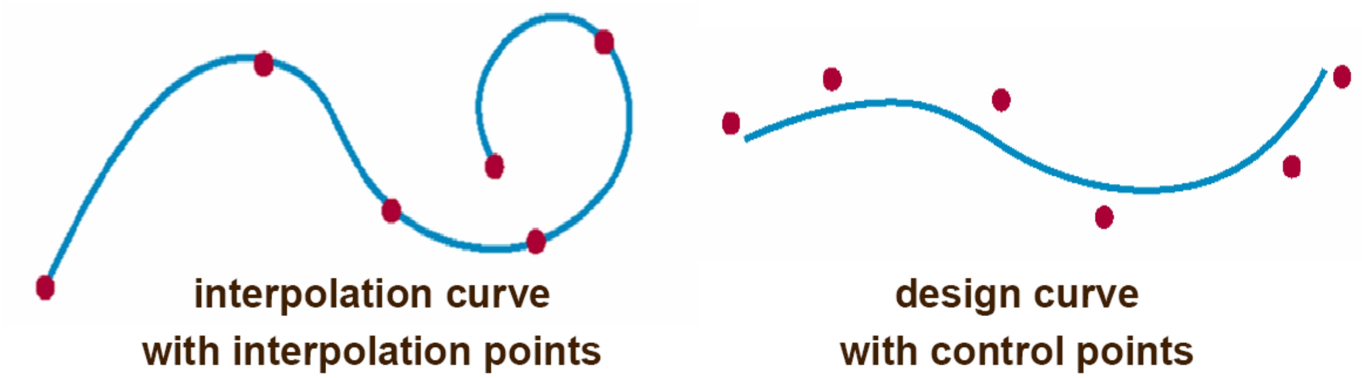
- An interpolation curve defines the exact position (point) that the curve must pass through, e.g. in a keyframe animation, an object must be at a particular point at a particular time.

- A design curve defines the general behaviour of the curve, e.g. what the curve should look like, and tuning the shape is often needed. The method is often used by designers.

Interpolation and design curves (2)

- The shape of the interpolation curve depends on the data points provided.

- The shape of the design curve depends on the control points, which do not lie on the curve, but allow adjustment of the shape by moving the points.



Design curves and local control

- The same approach is used in design curves.
- Each consists of separate, but joined parts.
- An important feature is local control.
- When a curve is designed, if one part is done, it would be preferred to keep its current shape when another part of the curve is adjusted.
- So the adjustment should influence only a small / local part of the curve – this is local control.

Local control

- Curves without local control
  - Natural splines
  - Bezier curves (if continuity enforced)

- Curves with local control
  - B-Splines
  - NURBS (Non-Uniform Rational B-Splines)

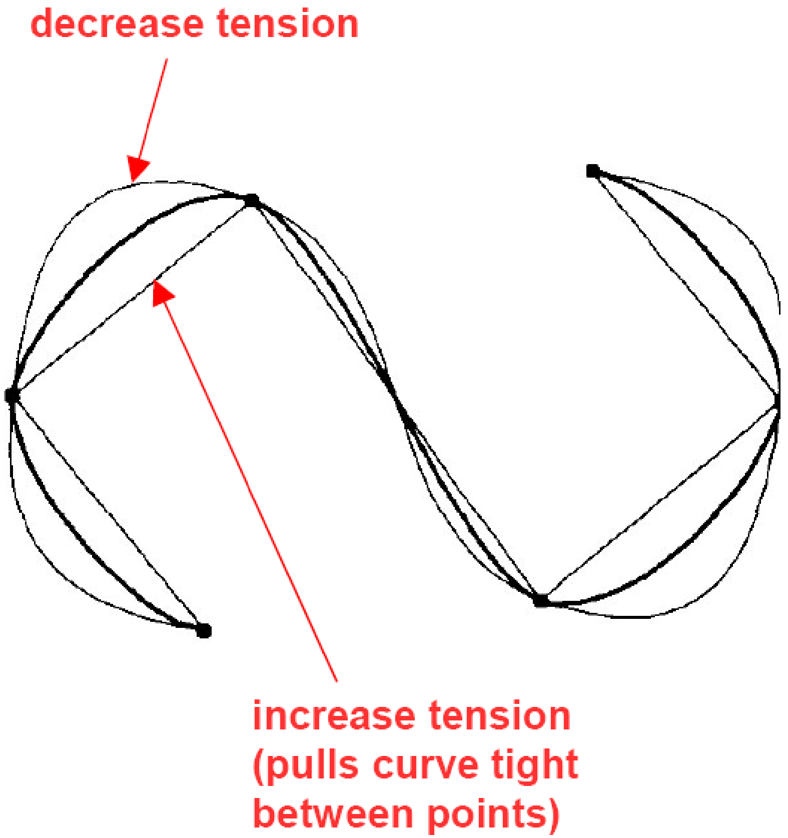
- A cubic curve with local control
  - Normally influenced by only 4 control points
  - which are the control points most local to it

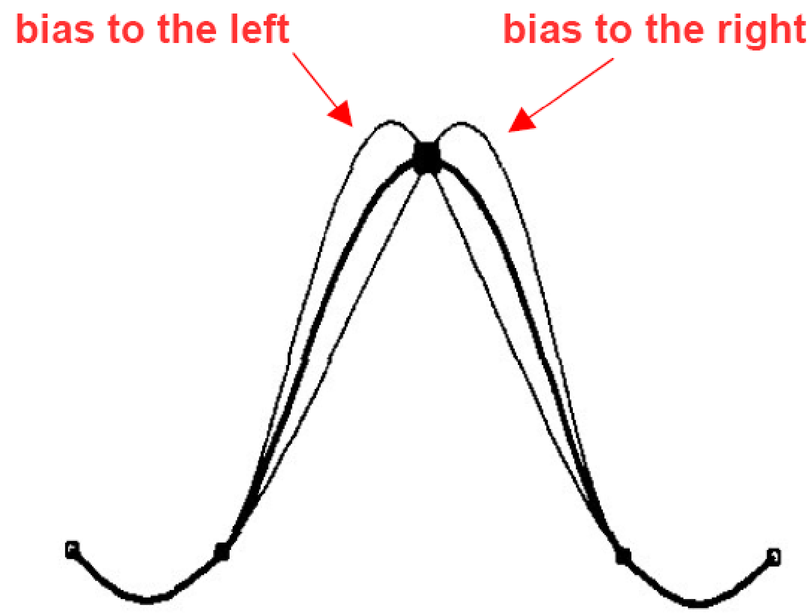
Pierre Etienne Bezier (1910-1999) – French engineer and mathematician with a long service at Renault, started his research in CADCAM in 1960 when he devoted a substantial amount of time on his UNISURF system. He focused on drawing machines, computer control, interactive free-form curve and surface design and 3D milling for manufacturing clay models and masters.

Forms of local control

- So far we have considered controlling the design curve by only moving the control points.
- Some types of curve provide further parameters to allow some control while keeping the control points fixed – important ones include tension and bias.
- Such control can apply to <u>both</u> interpolation and design curves.

Tension and bias





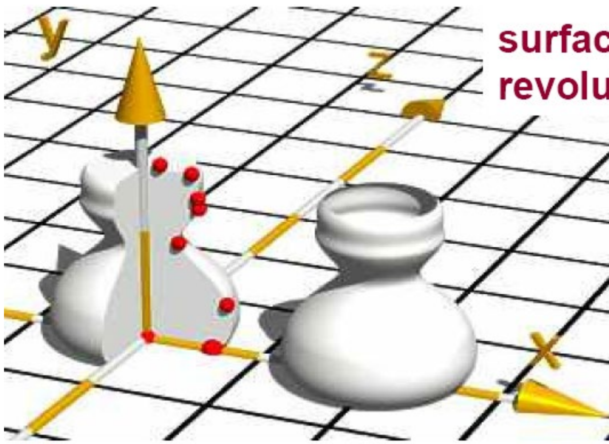
Types of parametric surface

- Revolved surface – a 2D curve is revolved around an axis, and the parameter is the rotation angle.

- Extruded surface – a 2D curve is moved perpendicular to its own plane, and the parameter is the straight-line depth.

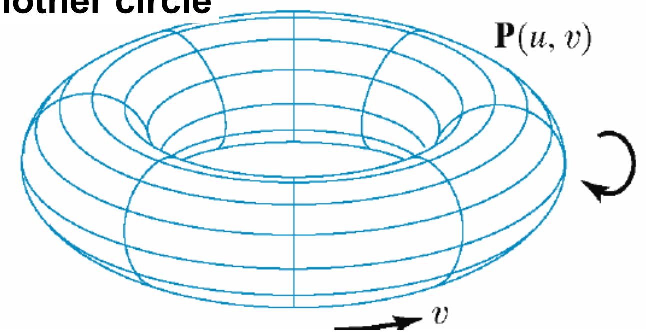
- Swept surface – a 2D curve is passed along a 3D path (which can be a curve), and the parameter is the path definition.

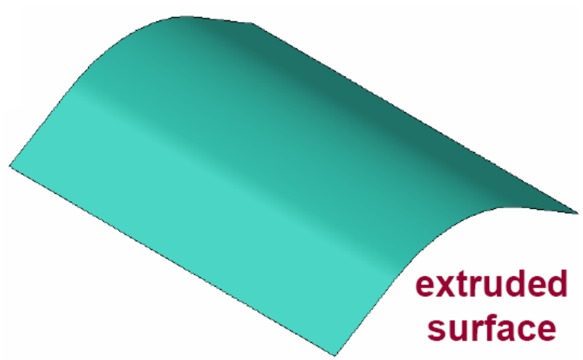
Examples of surface



surface of revolution

sweep surface produced by sweeping a circle along another circle

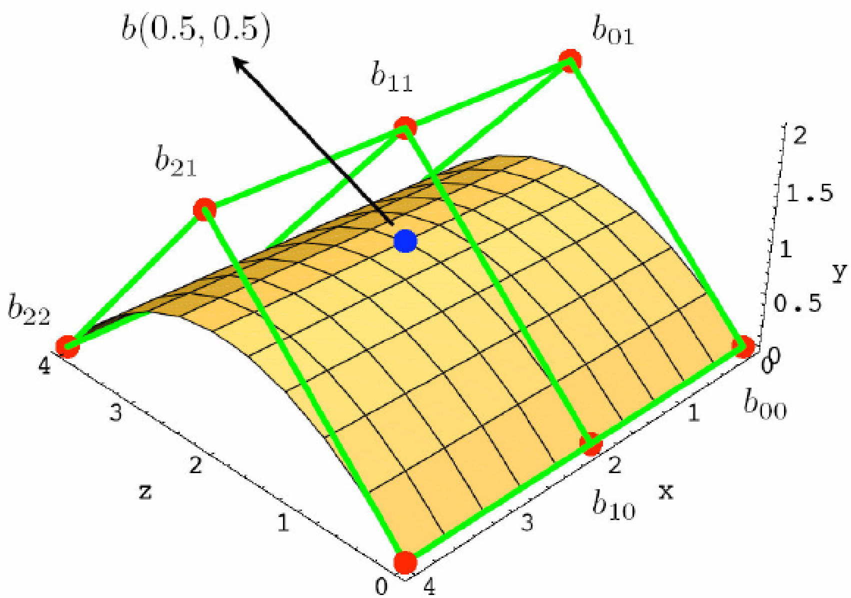




extruded surface

Tensor product surfaces

- They are the most widely used parametric surfaces.
- A tensor product surface combines two parametric curves (curves of curves such as the examples in the previous slide). Essentially these work in perpendicular directions.
- Parametric curves depend on 1 parameter (*t*) while parametric surfaces depend on 2 parameters (*u*,*v*).

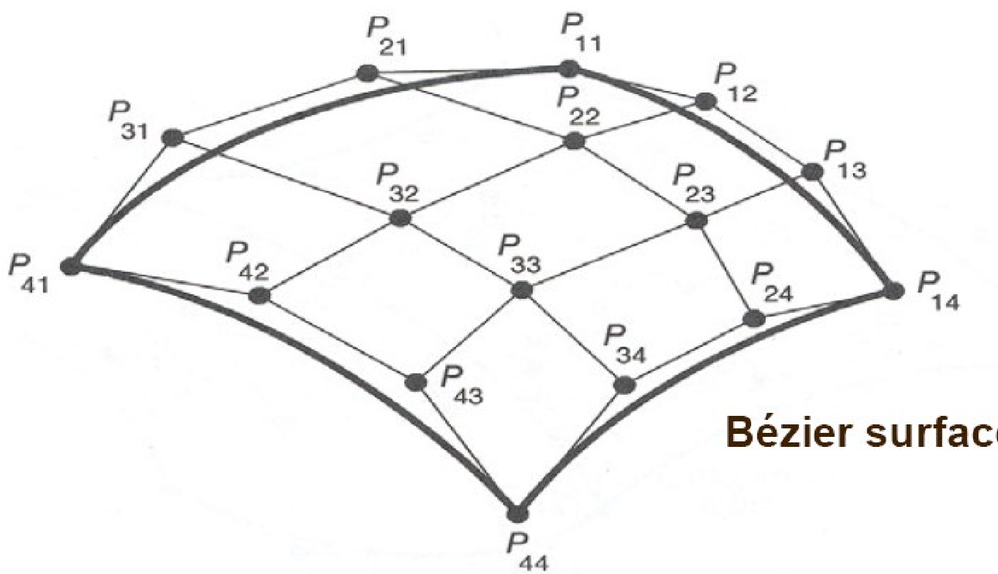


Interpolation and design surfaces

- As for curves, there are interpolation and design surfaces, and the design form is more common.
- There is a control grid, normally a rectangular array of control points.
- As the curve is broken into smaller curves, the surface is broken into surface patches.
- Local control becomes even more important.

Control grid for a surface patch

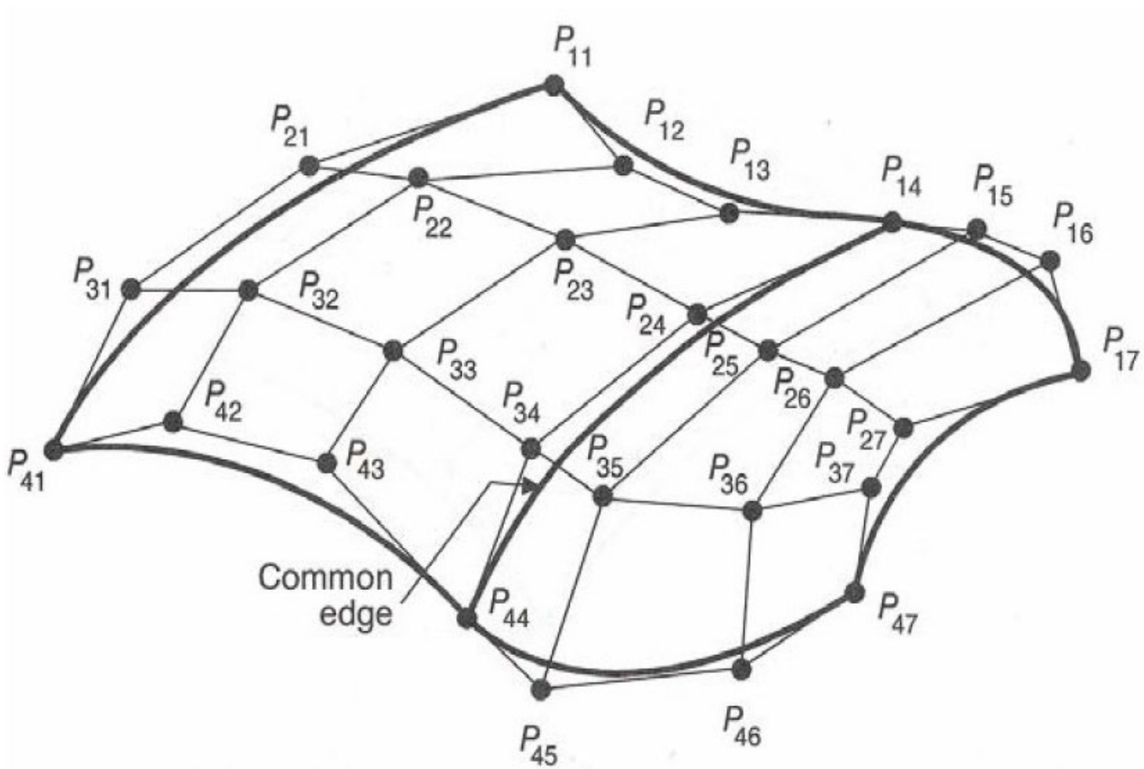
- In a cubic curve with local control, a curve segment is normally affected by only 4 control points.
- In a cubic surface with local control, a surface patch is affected by 16 control points, these being in a 4x4 grid.



Bézier surface patch

Joining patches

- The patches are joined, which is not always straightforward.
- Appropriate continuity at the boundaries must be ensured.

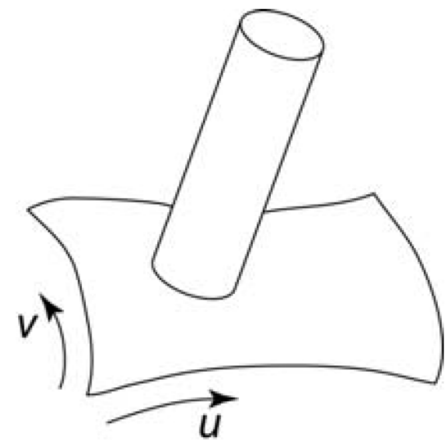


Trimming and control

➢ The control of the surface is performed as for curves

- the points in the control grid can be moved.

- some types of surface have tension and other parameters to use without having to move grid points.



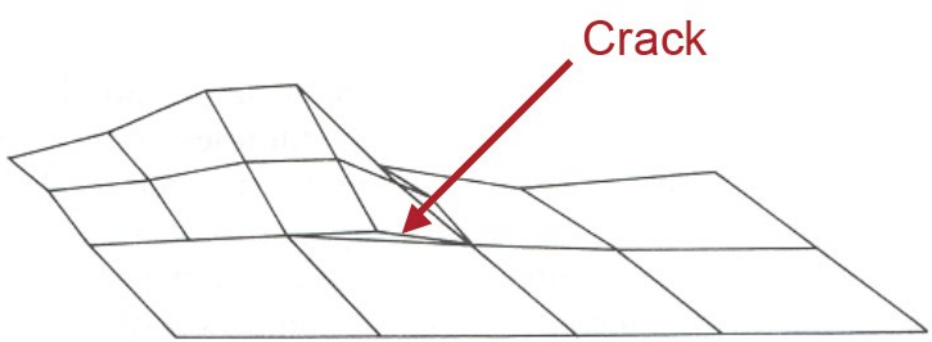
➢ If not all of the surface is to be displayed

- a trim line cutting the surface can be defined.

- all parts of the surface on one side of this line are removed.

Improving resolution

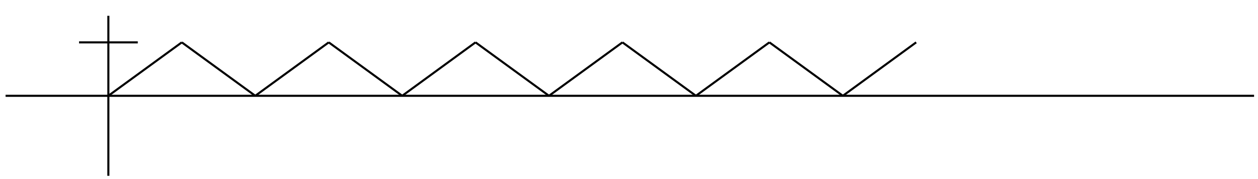
- To obtain finer detail, the number of patches can be increased.
- It is possible to do this adaptively, i.e. only increase the number of patches where extra detail is needed.
- This can lead to cracks in the surface unless care is taken.



Linear interpolation

Linear interpolation is useful (e.g. for animation) where *t* varies from 0 to 1 (and sometimes back again) over time.

```
float t = 0;
float dt = 0.01;
void onIdle(void)
{
  t = t+dt;
  if (t>1) {t = 1; dt = -0.01;}
  if (t<0) {t = 0; dt = 0.01;}
}
```



Parametric functions

```
// Draw a sinewave

   int i;
   float x, y;
   float d = 100.0;

   glBegin(GL_POINTS);
   for(i=0; i<=360; i=i+5)
   {
      x = (float)i;
      y = d*sin(i*(3.1416/180.0));
      glVertex2f(x,y);
   }
   glEnd();
```

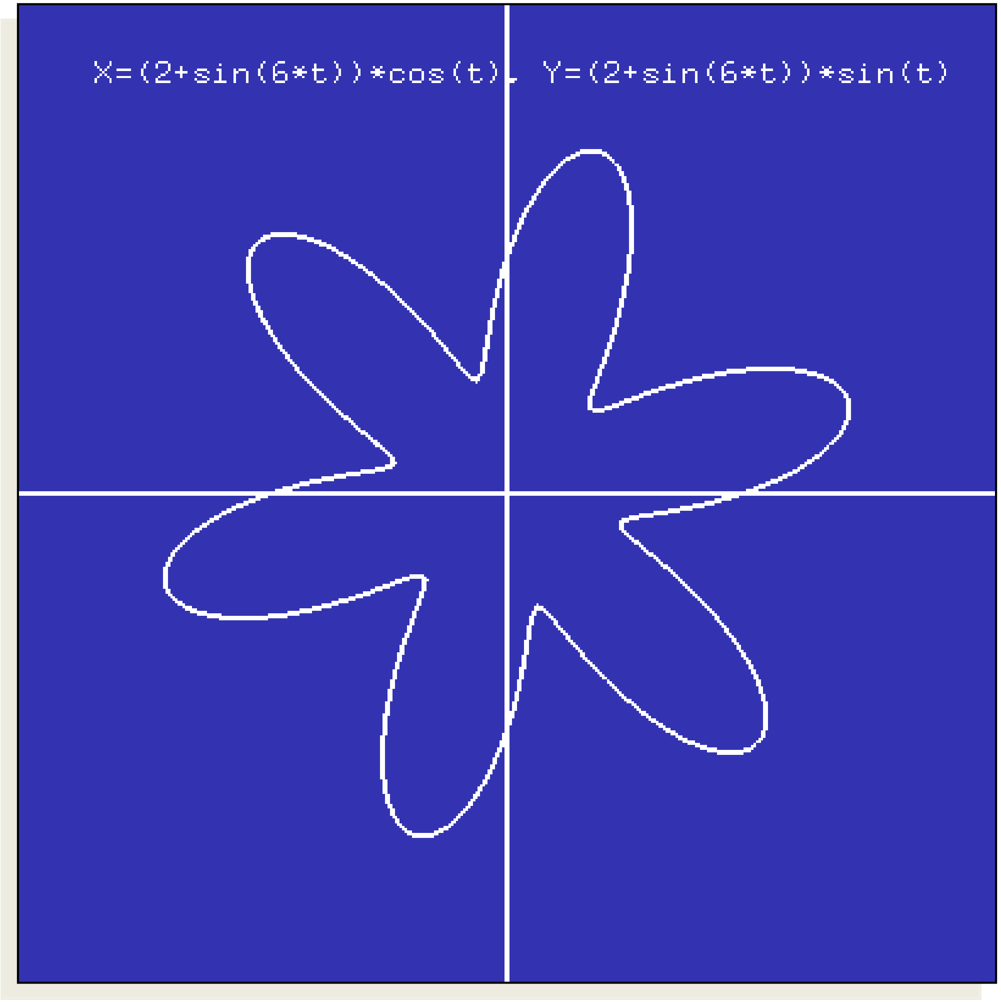
Parametric functions

```
// Draw a circle

   double x, y;
   double t;
   double r = 100;

   glBegin(GL_LINE_STRIP);
   for(t=0; t<=360; t+=1)
   {
      x = r*cos(t*3.1416/180);
      y = r*sin(t*3.1416/180);
      glVertex3f(x,y,0);
   }
   glEnd();
```

Parametric functions



Summary

- Parametric (i.e. tensor product) surfaces provide a flexible modelling tool.
- The number of patches required for a model is far fewer than that of polygons for a similar model.
- Some modelling systems are based on such surfaces (NURBS being the most popular).
- Using such models produces an additional computational load on rendering the image (hidden-surface removal, shading calculations, collision detection and so on).