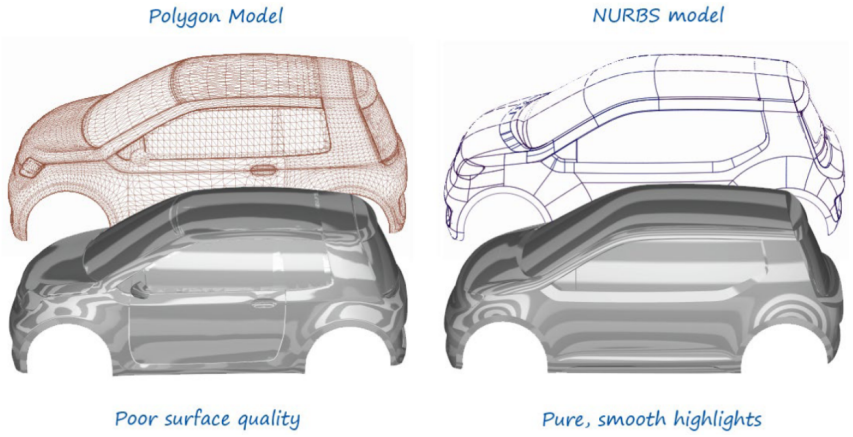
CPT205W05_参数曲线、曲面

必要性

为什么需要参数曲面?

为表示一个光滑的模型，相较于多边形组成的模型有棱有角（需要曲面细分改进），如果每个曲面曲线都是由光滑的参数方程控制（如：非均匀有理样条（Non uniform rational B-spline，*NURBS*）），那么就不会出现前者的问题。



为什么使用参数控制?

参数表面一般由两个独立变量组成。通过参数化（parameterisation）可以相对简单地表示自交或不可定向的（self-intersecting or non-orientable）曲面。这往往是隐式表达所无法表示的，即使隐函数存在，其镶嵌的表示？（tessellated representation）往往是不正确的。

使用参数控制的特点

参数曲线和曲面提供了一个灵活的建模工具，模型所需参数化表面的小块（patch）数量远小于使用多边形建模的数据量。

一些建模系统基于这样的表面（如NURBS）。

但使用这类模型渲染时会产生额外的计算负载（隐藏表面去除、阴影计算、碰撞检测等）。

参数曲线

在2D平面中的曲线定义为：

$$\begin{cases} x = x(t) \\ y = y(t) \end{cases} \quad where\ t \in [0,1]$$

每个t对应有且仅有一个点。当旋转时，其表达式不会改变

直线的表达式：

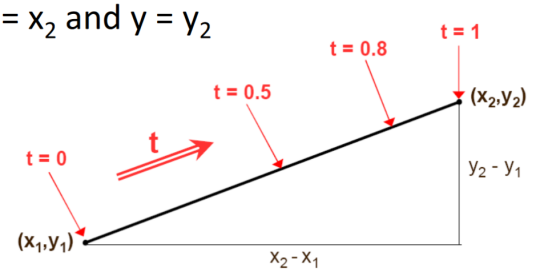
- Implicit representation: $y = a_0 + a_1 x$
- Parametric (explicit) representation:

  $x = x_1 + t(x_2 - x_1)$ $\quad (0 \le t \le 1)$

  $y = y_1 + t(y_2 - y_1)$

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

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



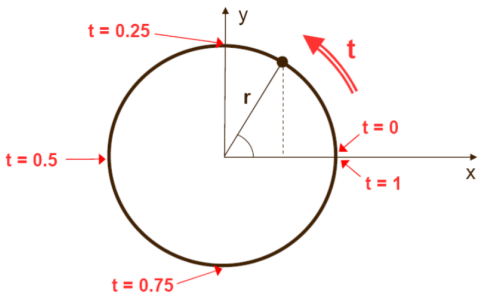
圆的表达式：

- Implicit representation:

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

- Parametric equation:

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



三次曲线的表达式：

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

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

应当使用何种曲线?

应当使用含有可快速计算函数的表达式（如sin, cos, exp, log等）

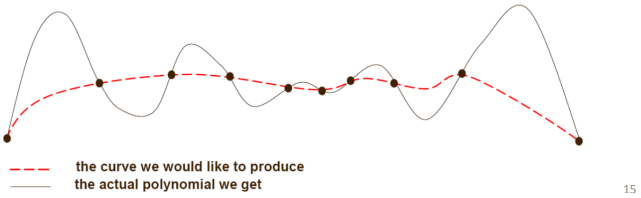
故多项式可表示为：$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, + \ldots + a_n t^n$

对于插值（interpolation），若有k个点，则应当选择n=k-1，从而找到正确的各个a_i值。

当k=2时，可以得到一条直线；当k=3时，可以得到一条抛物线.....

但当k很多时，可能会过拟合（雾

- 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.



低阶多项式曲线（Low-degree polynomial curves）

多项式应当高效地使用，阶数不宜过高。

若需要连接多个点，可以考虑将多个点分为多个包含少数点（如4个）的集合，再为每个集合分配低阶多项式曲线。这是样条曲线（spline）的基础。

样条曲线（spline）

样条曲线(Spline Curves)是指给定一组控制点而得到一条曲线，曲线的大致形状由这些点予以控制，一般可分为**插值样条（interpolation curve）**和**设计样条（design curve）**两种。

满足以下连续性：

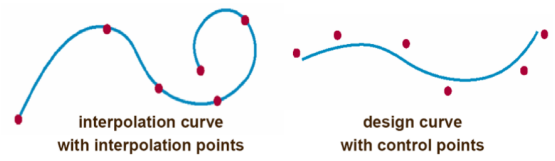
- 曲线连续（没有间断）continuity of the curve (no breaks)

- 切线（导数）连续（没有直的拐角）continuity of tangent (no sharp corners)
- （不一定满足）曲率的连续（防止一些光照问题）continuity of curvature (not essential but avoids some
  artefact from lighting)

满足以上要求至少需要三次曲线。

- **插值曲线（interpolation curve）**的形状取决于数据点（通过已知的离散数据点构造一个函数，使得该函数在给定的离散点上满足约束条件，即**插值函数必须经过所有的数据点。**）
- **设计曲线（design curve）**的形状取决于控制点（control points），控制点不在曲线上，但可通过移动控制点调整形状。

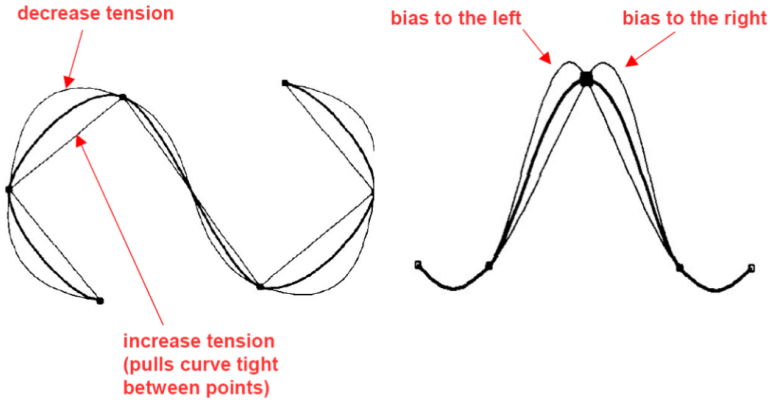
  每个部分都由单独但连接的部分组成。特色功能是局部控制（local control），如果曲线的一部分完成了，完成的部分会倾向于在调整其余部分时保持当前形状。即后来的调整只会影响曲线的一小部分。



局部控制：

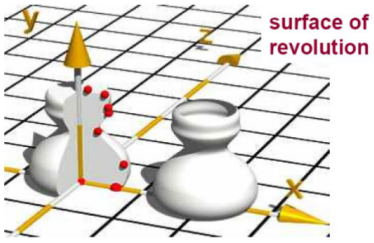
- 无局部控制的曲线：自然样条曲线（Natural splines），贝塞尔曲线（若强制其连续性）（Bezier curves (if continuity enforced)）
- 有局部控制的曲线：B 样条（B-Splines），NURBS（非均匀有理 B 样条）（NURBS (Non-Uniform Rational B-Splines)）
- 具有局部控制的三次曲线（cubic curve）：一般只受4个控制点影响，是最局部的控制点。

除了通过移动控制点控制设计曲线，还可以通过设置**张力（tension）**或**偏置（bias）**来控制曲线。对插值曲线和设计曲线都可使用。

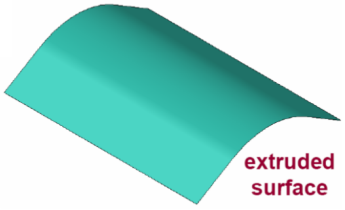


参数曲面

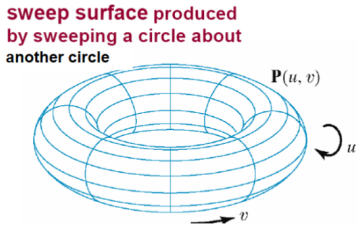
旋转曲面（Revolved surface）：2D曲线绕轴旋转，参数是旋转角度



挤出曲面（Extruded surface）：2D 曲线垂直于其自身的平面移动，参数是直线深度。



扫描曲面（Swept surface）：2D曲面现沿3D路径（如曲线）传递，参数是路径的定义。



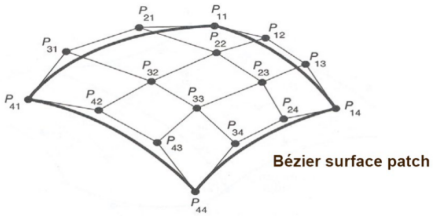
张量积曲面（Tensor product surface）：

应用最广泛的参数化曲面，结合了两条参数曲线，**分别在正交的方向上工作**。参数化曲面取决于两个参数(u,v)，而不是一个t。

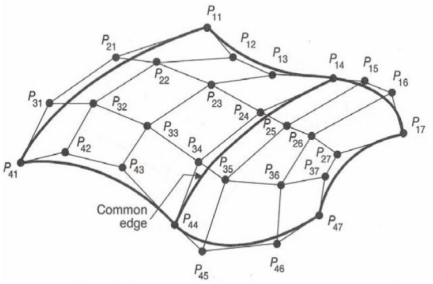
插值曲面和设计曲面：后者更为常见，使用控制网格（control grid），是控制点组成的矩阵。当曲线分解为更小的曲线时，曲面被裂解为曲面块（surface patches）。此时局部控制更加重要。

贝塞尔曲面：

有局部控制的三次曲线受4个控制点影响，有局部控制的立方体曲面受4*4个控制点影响。



连接曲面：必须确保边界的适当连续性。

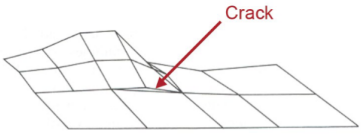


修建和控制（Trimming and control）：

控制方式类似控制曲面

若不想显示所有表面，可以定义切割曲面的切割线（trim line），删除此线一侧的表面的所有部分。

提高分辨率：可以增加patch的数量，可以自适应地执行此操作，即在需要额外细节的地方增加补丁数量。但需要注意，可能导致表面出现缝隙。



实战

线性插值（Linear interpolation）

对于动画很有用。

```
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;}
}
```

正弦波

```
// 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();
```

圆

```
// 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();
```