Computer Graphics

Curve and Surface Modeling

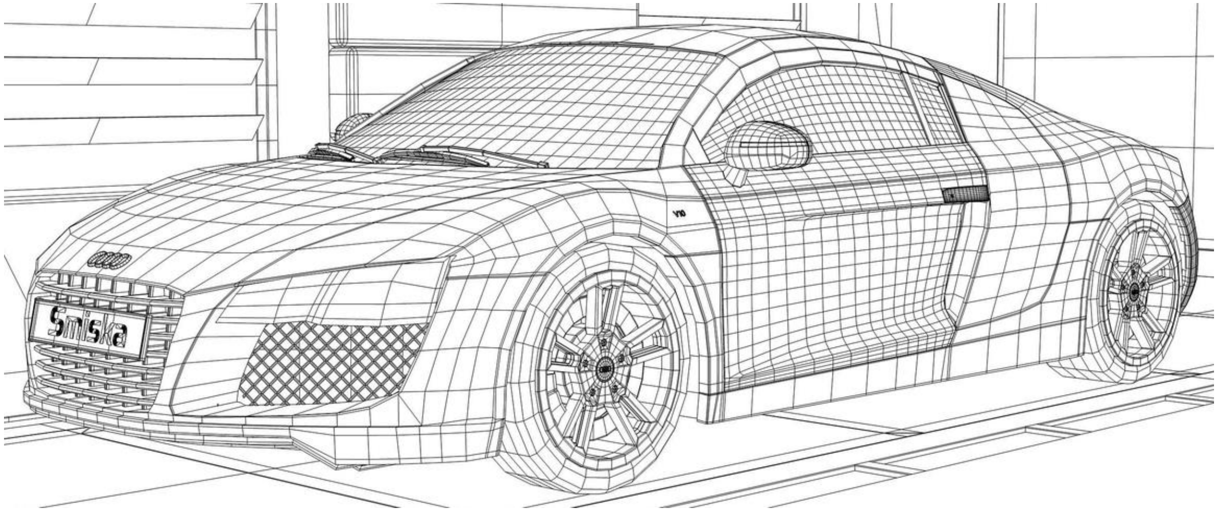
Teacher: A.Prof. Chengying Gao (高成英)

E-mail: mcsgcy@mail.sysu.edu.cn

School of Data and Computer Science

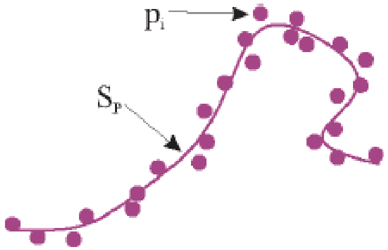
Outline

- Interpolation and Approximation
- Curve Modeling
  - Parametric curve
  - Cubic Hermite interpolation
  - Bézier curve
  - B-Spline
- Surface Modeling
  - Bézier surface

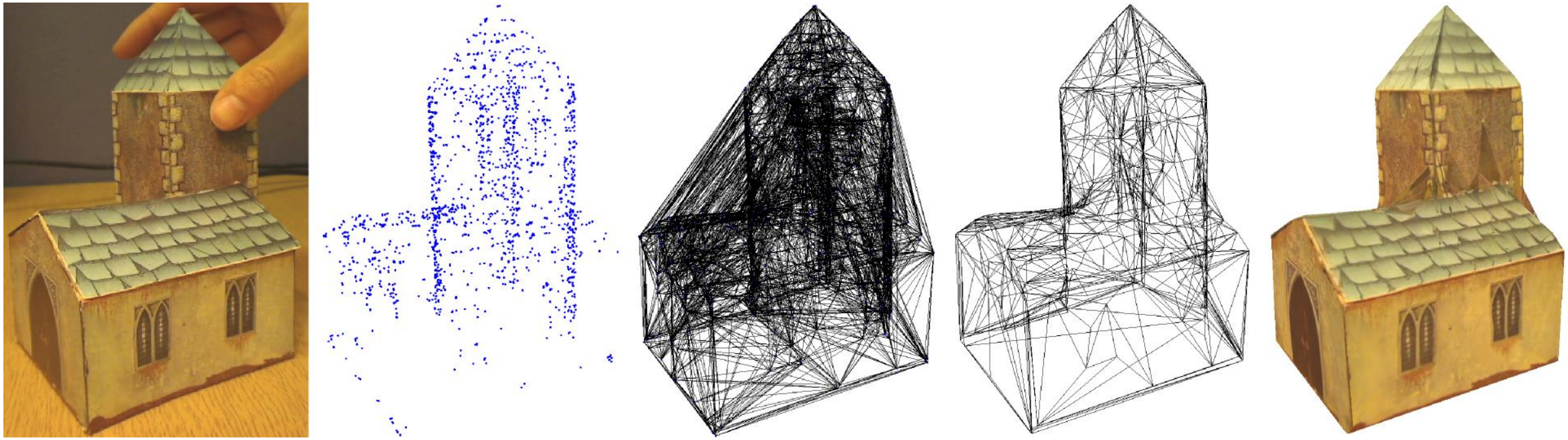


Introduction

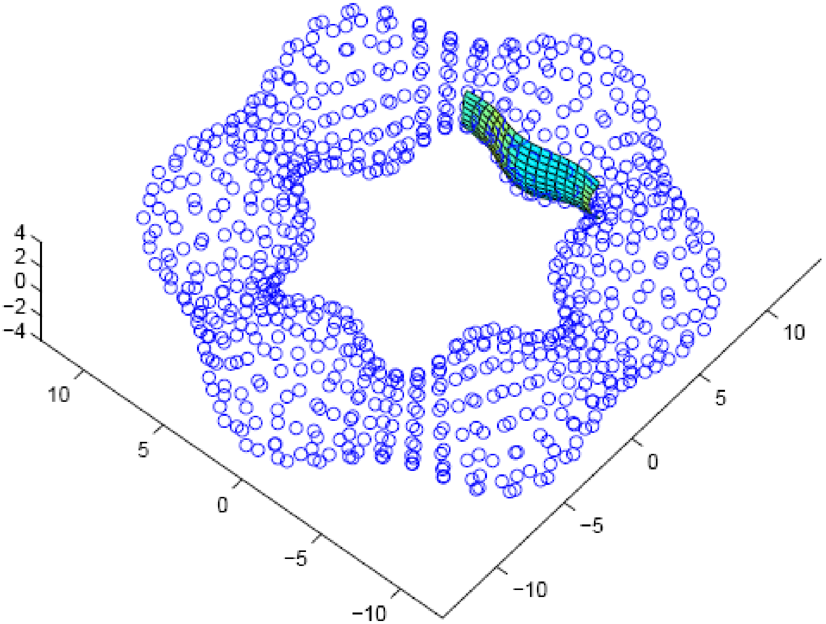
- Raw data is very popular in many experimental study and usually it need fitting before it can be understand well.



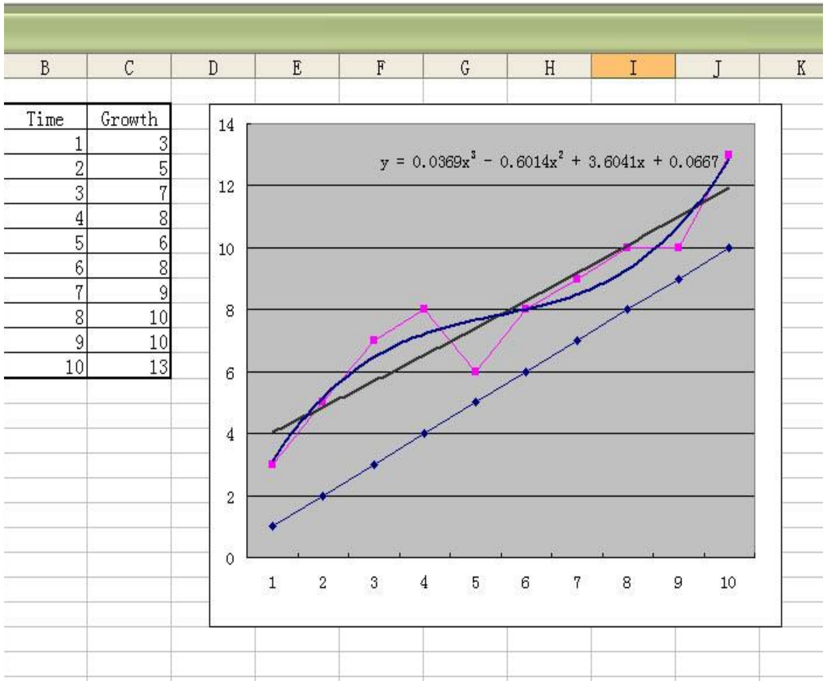
Introduction



Introduction



Surface fitting to 3D points



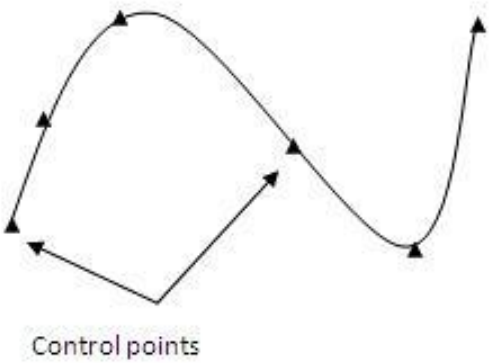
Time	Growth
1	3
2	5
3	7
4	8
5	6
6	8
7	9
8	10
9	10
10	13

$y = 0.0369x^3 - 0.6014x^2 + 3.6041x + 0.0667$

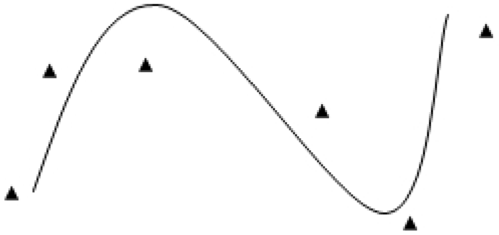
Chart by Microsoft Excel

Interpolation and approximation

- Interpolation: When the curve passes through all the control points then it is called as Interpolation.

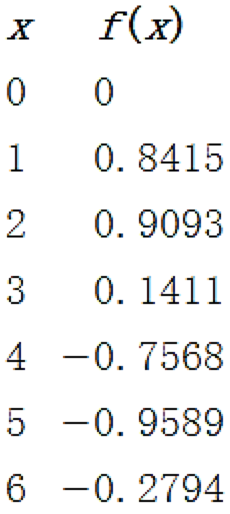


- Approximation: When the curve does not passes through the control points then it is called as approximation.

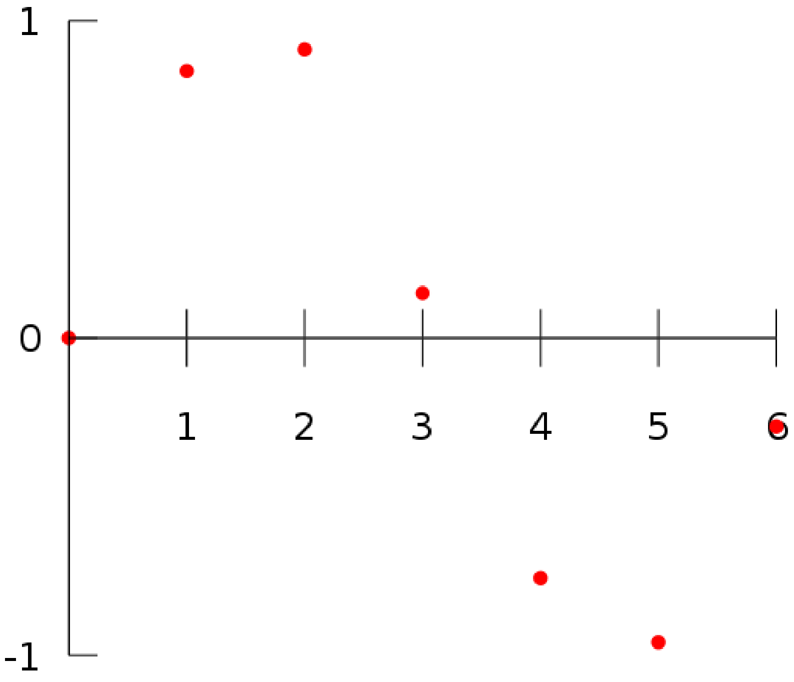


Interpolation case

- For example, suppose we have a table like this, which gives some values of an unknown function *f*.



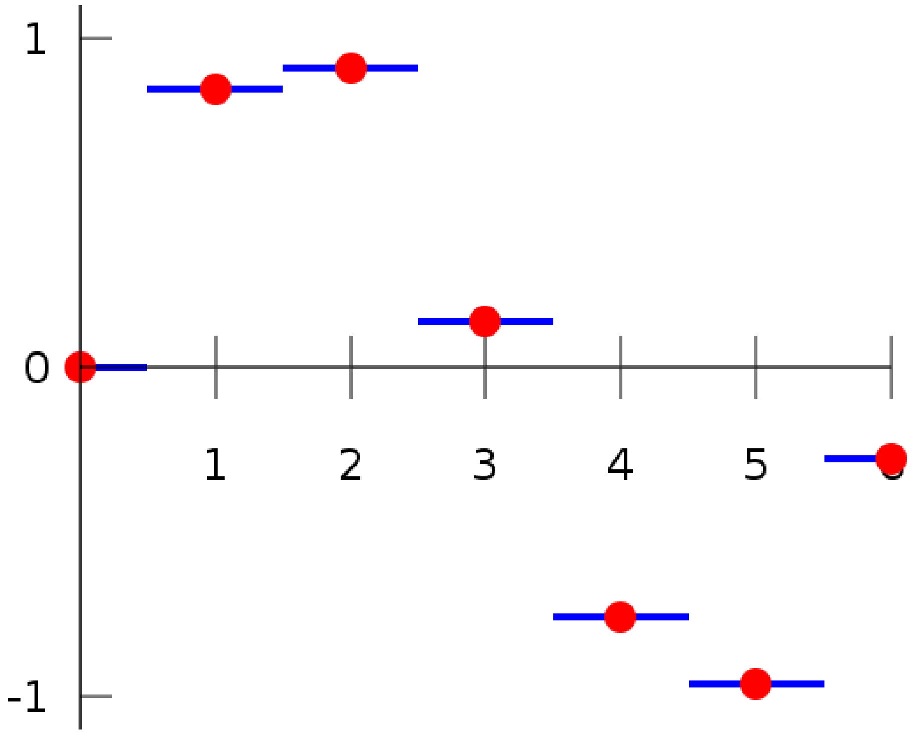
x	$f(x)$
0	0
1	0.8415
2	0.9093
3	0.1411
4	−0.7568
5	−0.9589
6	−0.2794



Interpolation provides a means of estimating the function at intermediate points, such as x = 2.5.

Piecewise constant interpolation

- The simplest interpolation method is to locate the nearest data value, and assign the same value.



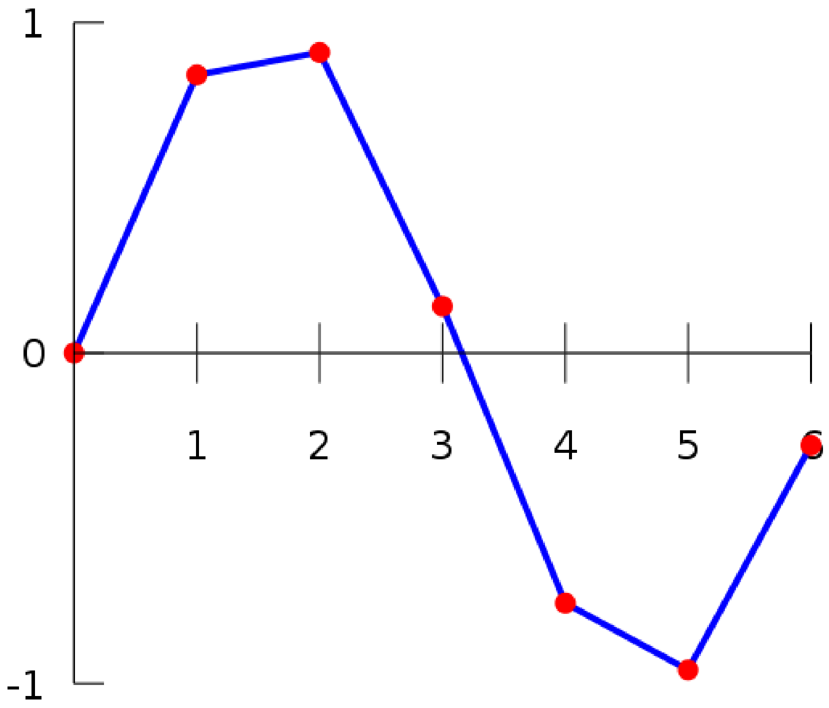
Linear interpolation

- Generally, linear interpolation takes two data points, say (x_a,y_a) and (x_b,y_b), and the interpolant is given by:

$$y = y_a + (y_b - y_a)\frac{x - x_a}{x_b - x_a} \text{ at the point } (x, y)$$

$$\frac{y - y_a}{y_b - y_a} = \frac{x - x_a}{x_b - x_a}$$

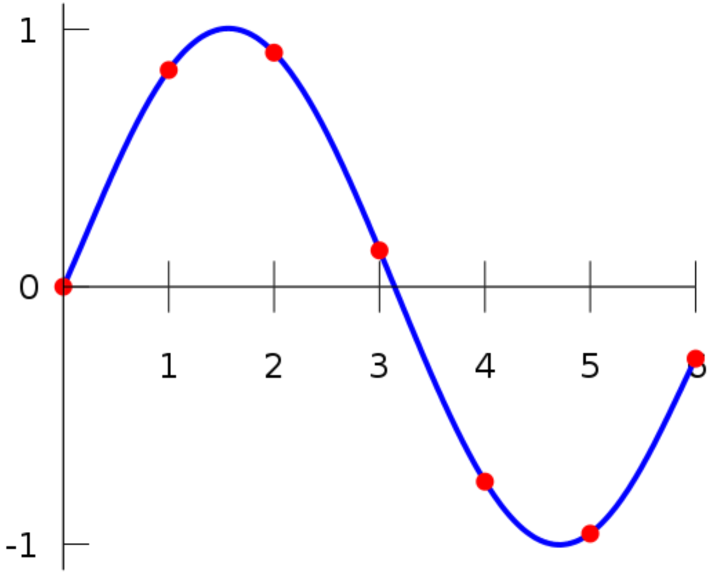
$$\frac{y - y_a}{x - x_a} = \frac{y_b - y_a}{x_b - x_a}$$



Polynomial interpolation

- Polynomial interpolation is a generalization of linear interpolation. Note that the linear interpolation is a linear function. We now replace this interpolation with a polynomial of higher degree.
- The following sixth degree polynomial goes through all the seven points:

$$
\begin{aligned}
f(x) = &-0.0001521x^6 + 0.003130x^5 \\
&+0.07321x^4 - 0.3577x^3 \\
&+0.2255x^2 + 0.9038x
\end{aligned}
$$

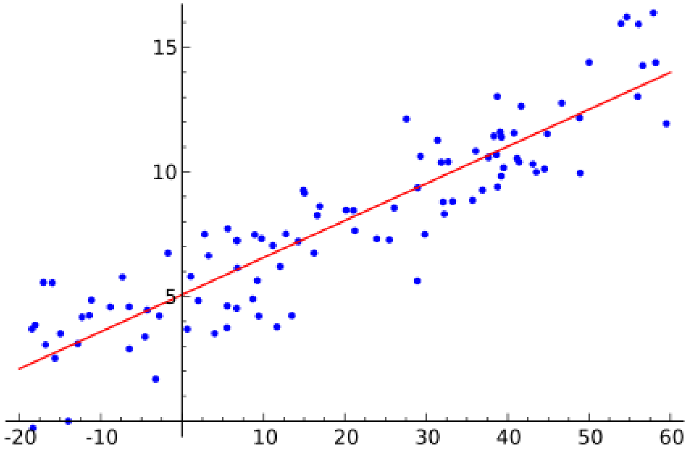


Approximation – Least squares fitting

- Linear least squares
  - A fitting model is a linear one when the model comprises a linear combination of the parameters, i.e.,

$$f(x,\beta)=\sum_{j=1}^{m}\beta_j\phi_j(x),$$

where the function ϕ_j is a function of x .



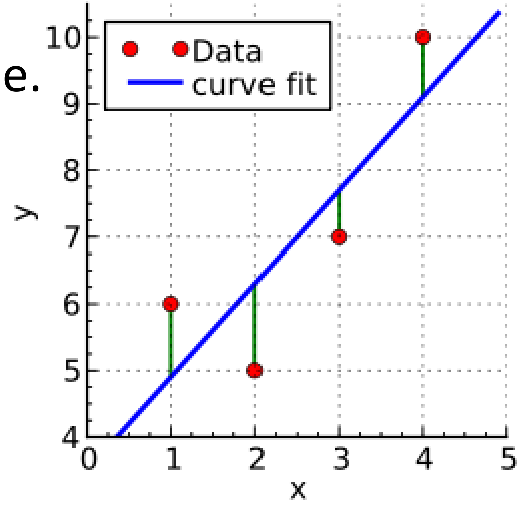
Least squares fitting example

- As a result of an experiment, four (x, y) data points were obtained, (1, 6), (2, 5), (3, 7), and (4, 10).

- We hope to find a line $y = \beta_1 + \beta_2 x$ that best fits these four points. In other words, we would like to find the numbers β_1 and β_2 that approximately solve the over-determined linear system

$$\beta_1 + 1\beta_2 = 6, \quad \beta_1 + 2\beta_2 = 5,$$
$$\beta_1 + 3\beta_2 = 7, \quad \beta_1 + 4\beta_2 = 10.$$

of four equations in two unknowns in some "best" sense.



- A **residual** is defined as the difference between the actual value of the dependent variable and the value predicted by the model.

$$r_i = y_i - f(x_i, \beta).$$

Least squares fitting example

- The "error", at each point, between the curve fit and the data is the difference between the right- and left-hand sides of the equations above. The least squares approach to solving this problem is to try to make the sum of the squares of these errors as small as possible; that is, to find the minimum of the function

$$
\begin{aligned}
S(\beta_1,\beta_2) &= [6-(\beta_1+1\beta_2)]^2+[5-(\beta_1+2\beta_2)]^2 \\
&\quad +[7-(\beta_1+3\beta_2)]^2+[10-(\beta_1+4\beta_2)]^2 \\
&= 4\beta_1^2+30\beta_2^2+20\beta_1\beta_2-56\beta_1-154\beta_2+210
\end{aligned}
$$

- The minimum is determined by calculating the partial derivatives of $S(\beta_1,\beta_2)$ with respect to β_1 and β_2 and setting them to zero

$$\frac{\partial S}{\partial \beta_1}=0=8\beta_1+20\beta_2-56,\quad \frac{\partial S}{\partial \beta_2}=0=20\beta_1+60\beta_2-154.$$

- This results in a system of two equations in two unknowns, called the normal equations, which give, when solved $\beta_1=3.5, \beta_2=1.4$,

and the equation $y=3.5+1.4x$ of the line of best fit.

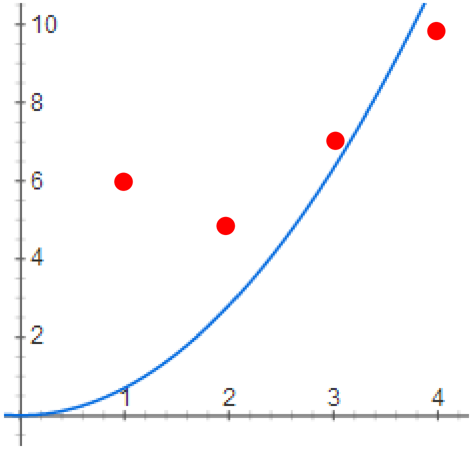
Approximation – Quadratic least squares fitting

- Importantly, in linear least squares, we are not restricted to using a line as the model as in the above example. For instance, we could have chosen the restricted quadratic model $y = \beta_1 x^2$. This model is still linear in the β_1 parameter, so we can still perform the same analysis, constructing a system of equations from the data points: $6 = \beta_1(1)^2$, $5 = \beta_1(2)^2$ $7 = \beta_1(3)^2$, $10 = \beta_1(4)^2$

- The partial derivatives with respect to the parameters (this time there is only one) are again computed and set to 0:

$$\frac{\partial S}{\partial \beta_1} = 0 = 708\beta_1 - 498$$

and solved $\beta_1 = 0.703x^2$



- leading to the resulting best fit model $y = 0.703x^2$

General - Least Square Methods

- How to draw a curve approximately fitting to raw data?
  - Raw data usually has noise. The values of dependent variables vary even though all the independent variables are constant. Therefore, the estimation of the trend the dependent variables is needed. This process is called regression or curve fitting.
  - The estimated equation (matrix) satisfy the raw data. However, the equation is not usually unique, and the equation or curve with a minimal deviation from all data points is desirable.
  - This desirable best-fitting equation can be obtained by least square method which uses the minimal sum of the deviations squared (偏差的平方和) from a given set of data.

Least square formulation

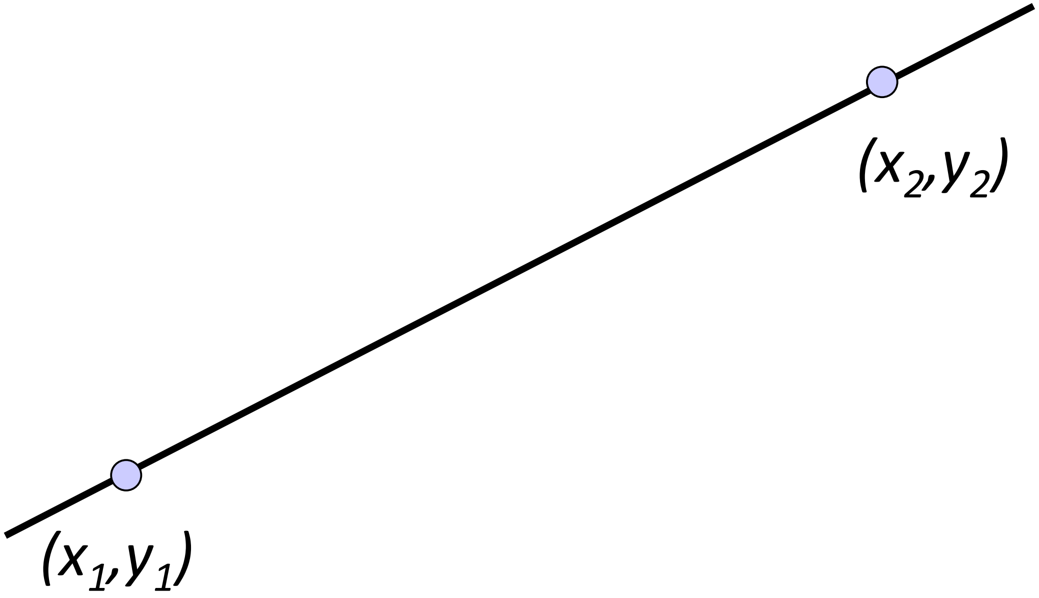
- If you have a data set *(x₁,y₁), (x₂, y₂) , … , (xₙ, yₙ)* and the best curve *f(x)* should be with the property as follows

Minimum Least Square error

$$E = \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

Least square line

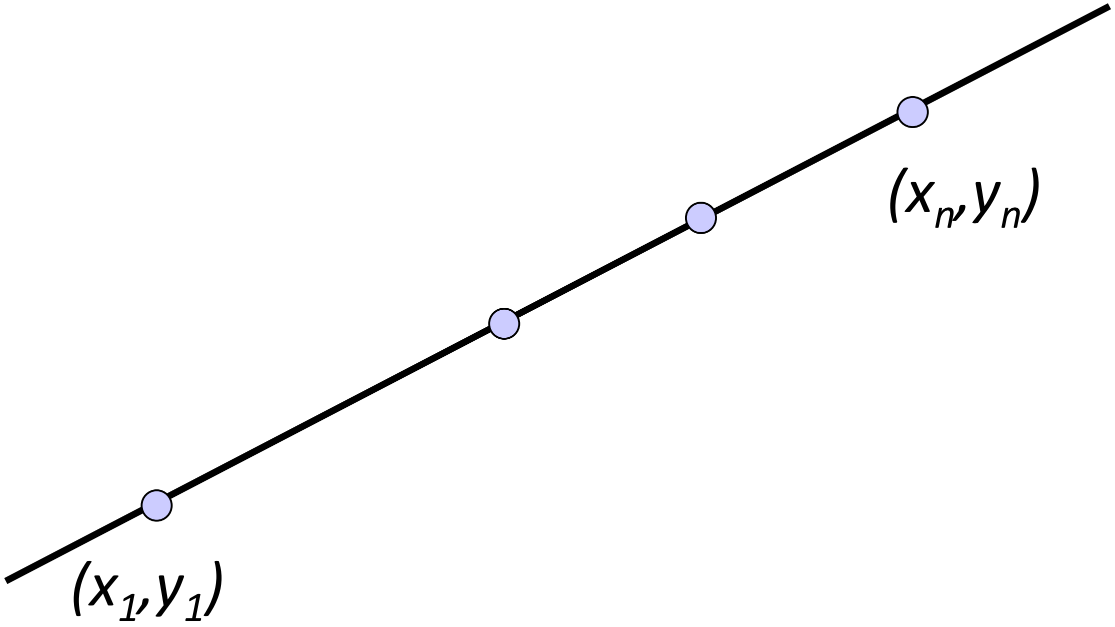
- When $n = 2$, $E = 0$



Least square line

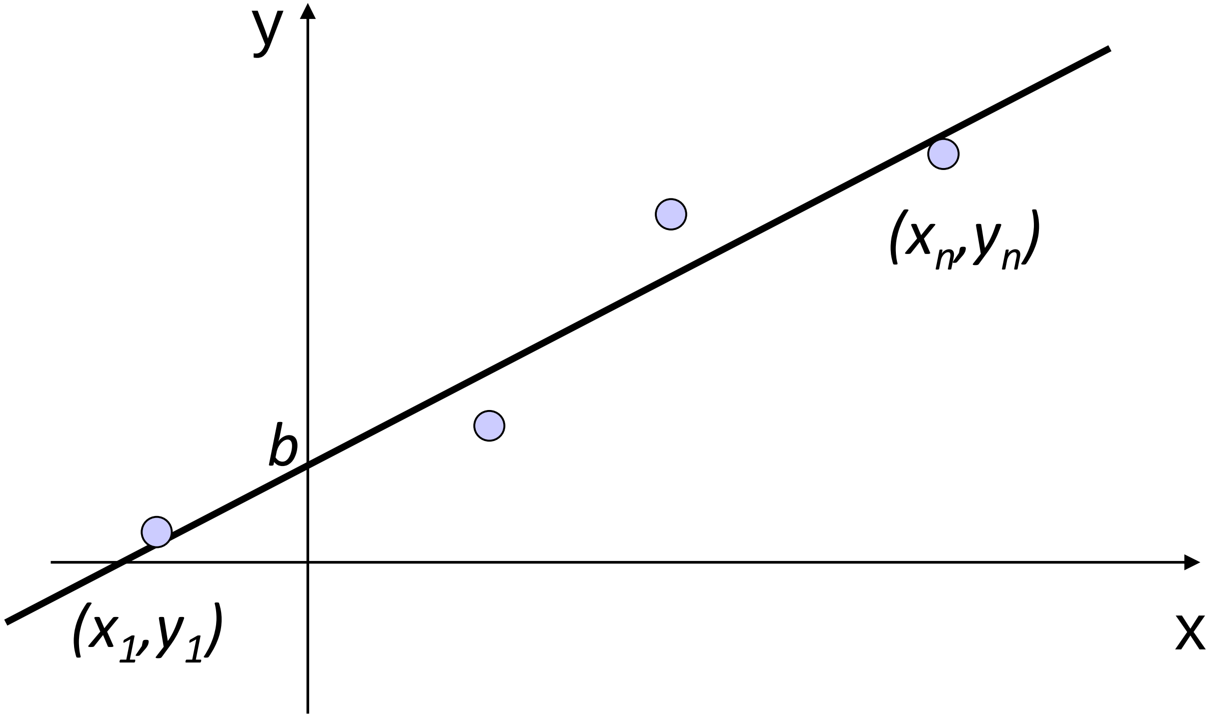
- When $n>2$, if $(x_1,y_1), (x_2, y_2)$, … , (x_n, y_n) are collinear,

  $E = 0$



Least square line

- Line equation $y = kx + b$



Least square line

- How to get *k* and *b* ?

$$\begin{cases} kx_1 + b = y_1 \\ kx_2 + b = y_2 \\ \cdots \\ kx_n + b = y_n \end{cases} \quad \text{or} \quad \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & 1 \\ x_n & 1 \end{bmatrix} \begin{bmatrix} k \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Mostly an approximation solution can exist, when the rank of the coefficient matrix is 2, which is the column number.

Least square line

- How to get *k* and *b* ?

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} k \\ b \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix} \begin{bmatrix} k \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

Least square line

- How to get *k* and *b* ?

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix} \begin{bmatrix} k \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

The unique solution of this system *k* and *b* can satisfy the following condition and a least square line is obtained.

Minimum $E = \sum_{i=1}^{n} (f(x_i) - y_i)^2$

Least square line

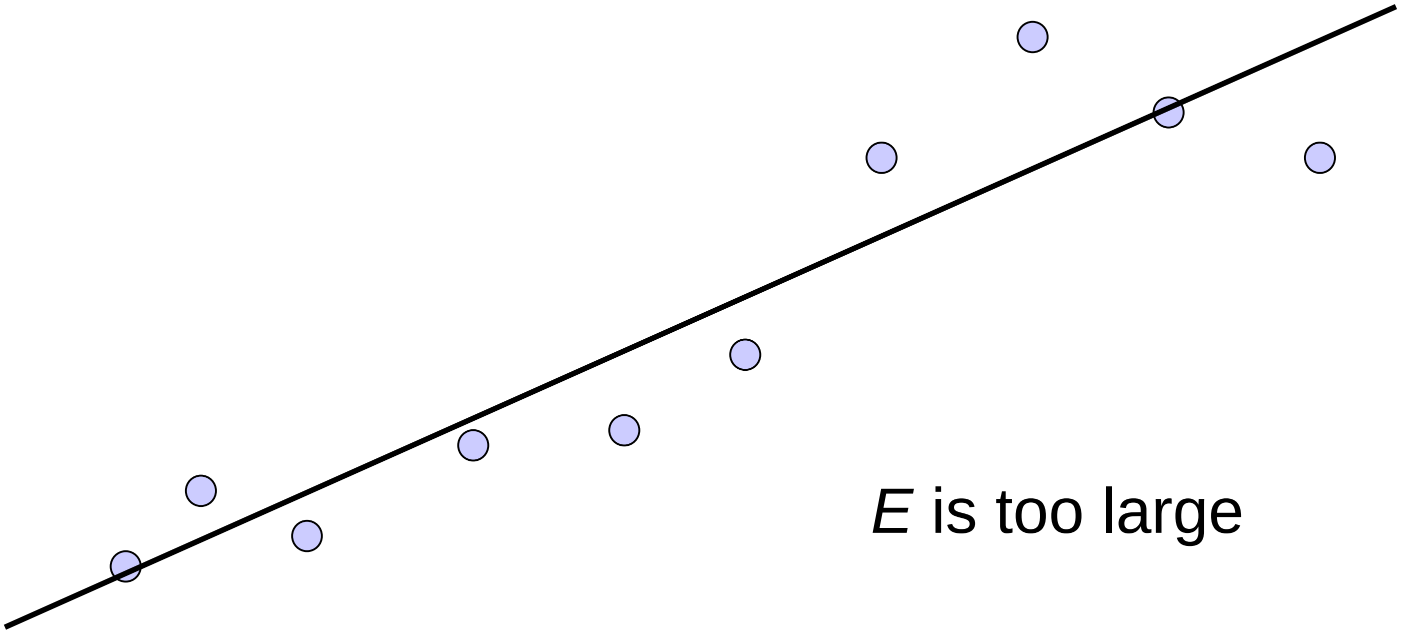
- For parametric definition, the least square line problem is now to find a_0, a_1, b_0, b_1 satisfying

$$\text{Minimum } E = \sum_{i=1}^{n} \left[(x(t_i) - x_i)^2 + (y(t_i) - y_i)^2 \right]$$

- Ways to choose $t_1, t_2, \cdots, t_n$ will affect the result.

Least square curve

- Why we need least square curve? When raw data is too complicated, least square line is not good enough.



E is too large

Least square curve

- In more general, $f(x) \in P_k$ can be a polynomial of degree k

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$$

- The problem becomes now to find $a_0, a_1, \cdots, a_k$ satisfying the following

Minimum $E = \sum_{i=1}^{n} (f(x_i) - y_i)^2$

Least square curve

- Systems to be solved are

$$
\begin{aligned}
a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_k x_1^k &= y_1 \\
a_0 + a_1 x_2 + a_2 x_2^2 + \cdots + a_k x_2^k &= y_2 \\
\cdots \qquad \cdots & \\
a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_k x_n^k &= y_n
\end{aligned}
$$

And

$$
\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}
$$

The coefficient matrix $\boldsymbol{M}$： $\boldsymbol{n\ x\ (k+1)}$

Least square curve

- When $n > k$ and rank of M is $k+1$, we solve the following system to get the least square curve

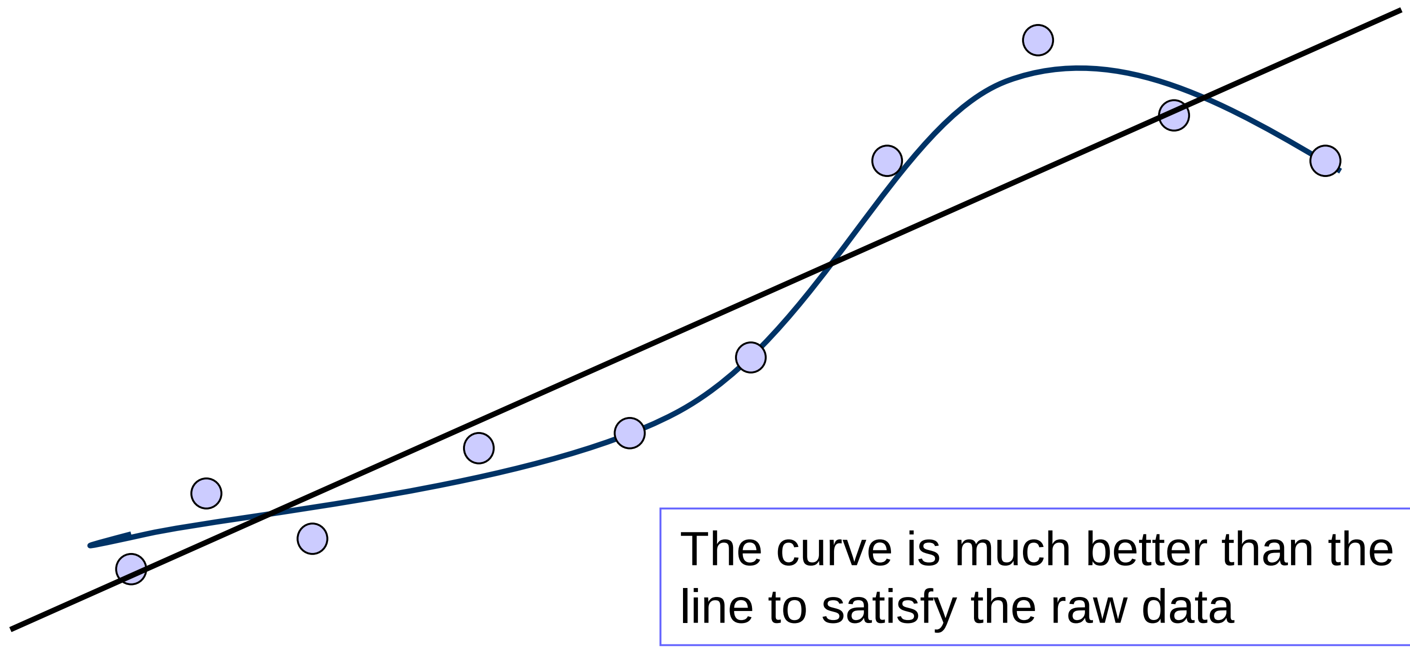
$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & x_3^k & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^k & x_2^k & x_3^k & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$(M^T M)X = M^T D$$

The coefficient matrix $\boldsymbol{M^T\ M}$: $(\boldsymbol{k}+\mathbf{1})\ \boldsymbol{x}\ (\boldsymbol{k}+\mathbf{1})$

Least square curve

- If $x_1 \neq x_2 \neq \cdots \neq x_n$ and $n > k$, we can always find the unique solution of the system, and it will be the least square solution to the original system.

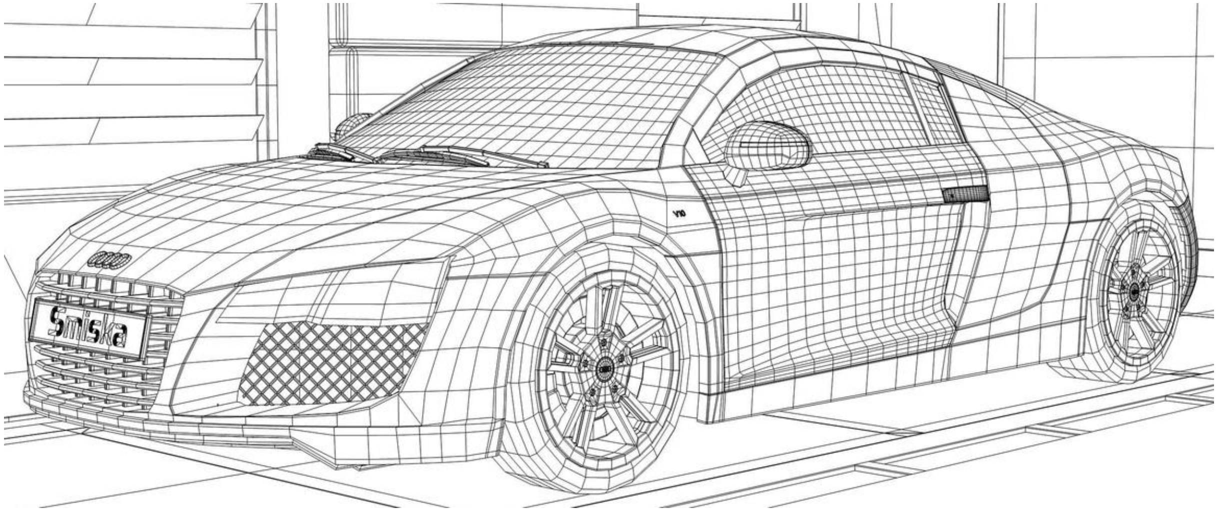


The curve is much better than the line to satisfy the raw data

Outline

- Interpolation and Approximation
- Curve Modeling
  - Parametric curve
  - Cubic Hermite interpolation
  - Bézier curve
  - B-Spline

- Surface Modeling
  - Bézier surface



Classification of curves

$$y = x^2 + 5x + 3 \longrightarrow y = f(x)$$

(explicit curve)

$$(x - x_c)^2 + (y - y_c)^2 - r^2 = 0 \longrightarrow g(x,y) = 0$$

(implicit curve)

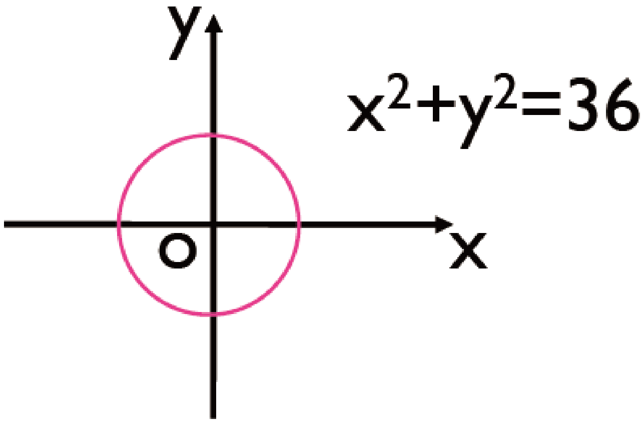
$$\begin{aligned} x &= x_c + r \cdot \cos\theta \\ y &= y_c + r \cdot \sin\theta \end{aligned} \longrightarrow \begin{cases} x = x(t) \\ y = y(t) \end{cases}$$

(parametric curve)

Classification of curves

implicit curve

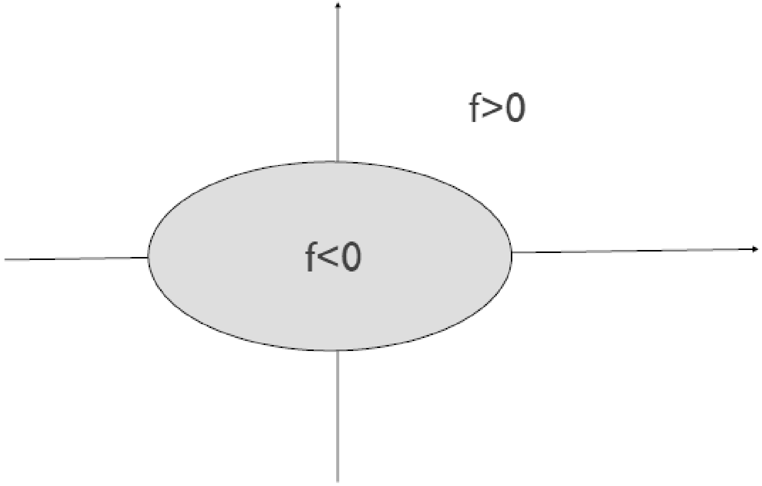
- planar: $f(x,y)=0$: $x^2+y^2-36=0$
- 3D curves

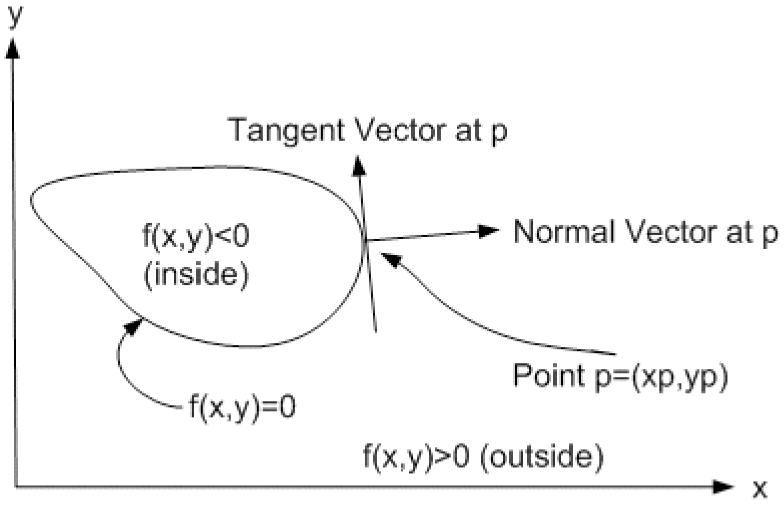


$$\begin{cases} f(x,y,z)=0, \\ g(x,y,z)=0. \end{cases}$$

Implicit curves

- **Advantage** of implicit curve:
  - To a point (x,y), it is easy to detect whether f(x,y) is >0 ,<0 or =0.
- **Disadvantage** of implicit curve:
  - To a curve f(x,y) = 0, it is difficult to find the point on it.



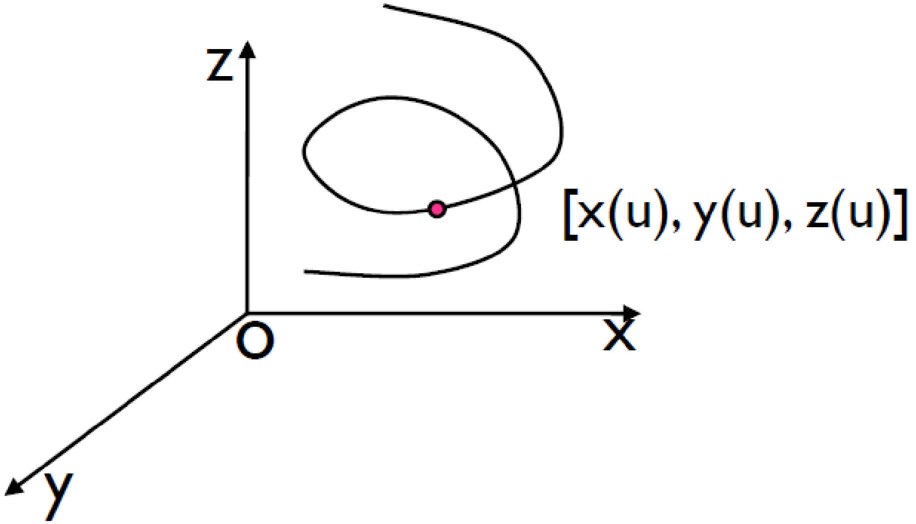


Parametric curves

- Variable is a scalar, and function is a vector:

$$C = C(u) = [x(u), y(u), z(u)],$$

- Every element of the vector is a function of the variable (the parameter)



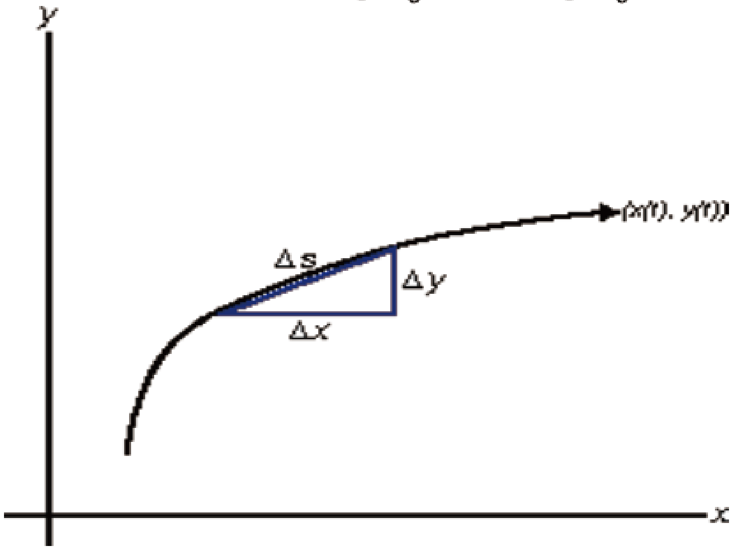
Parametric curves

given a curve **C**(u), its tangent is ***T***=***C***'(u).

difference of arc length:

$$(ds)^2=(dx)^2+(dy)^2+(dz)^2=((x')^2+(y')^2+(z')^2)d^2u$$

- Arc length: $s=\int_{u_0}^{u} ds=\int_{u_0}^{u}\sqrt{(x')^2+(y')^2+(z')^2}\,du$



Least square parametric curve

- Parametric definition of the curve (3D)

$$\begin{cases} x(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_k t^k \\ y(t) = b_0 + b_1 t + b_2 t^2 + \cdots + b_k t^k \\ z(t) = c_0 + c_1 t + c_2 t^2 + \cdots + c_k t^k \end{cases}$$

- Square Error

$$E = \sum_{i=1}^{n} \left[(x(t_i) - x_i)^2 + (y(t_i) - y_i)^2 + (z(t_i) - z_i)^2 \right]$$

Least square curve – general case

- General method to solve the problem is based on the following

$$\frac{\partial E}{\partial a_i} = 0, \frac{\partial E}{\partial b_i} = 0, \frac{\partial E}{\partial c_i} = 0, i = 0, \cdots, k$$

- The least square solution can be got by solving a related linear system

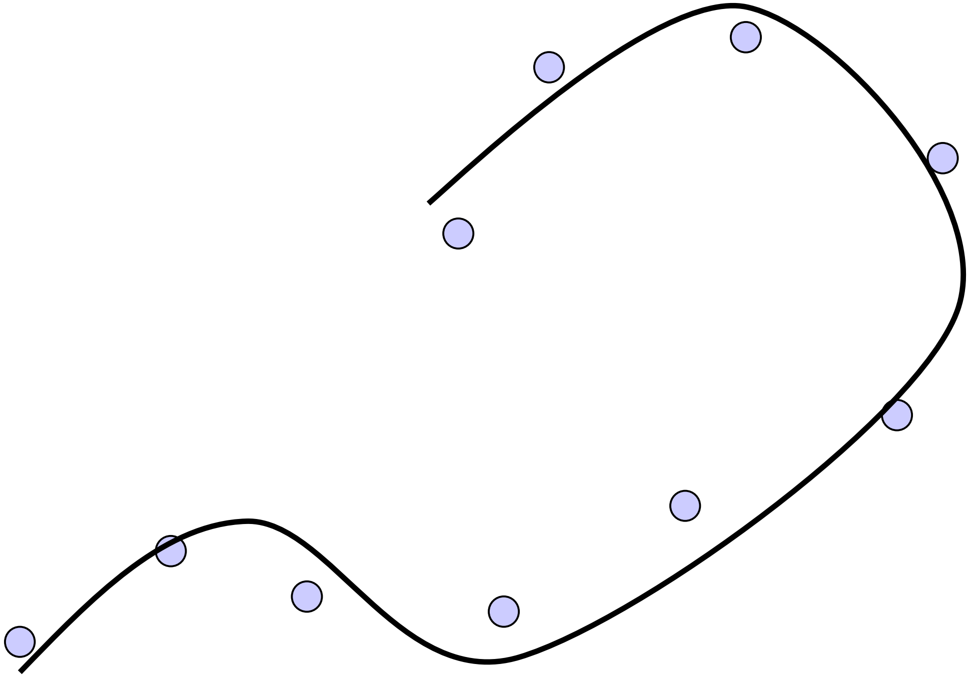
Least square parametric curve

- *Remark: the different choice of t_1 , t_2, …, t_n will lead different result.*
- Chord length (弦长) parameter is one of the best.

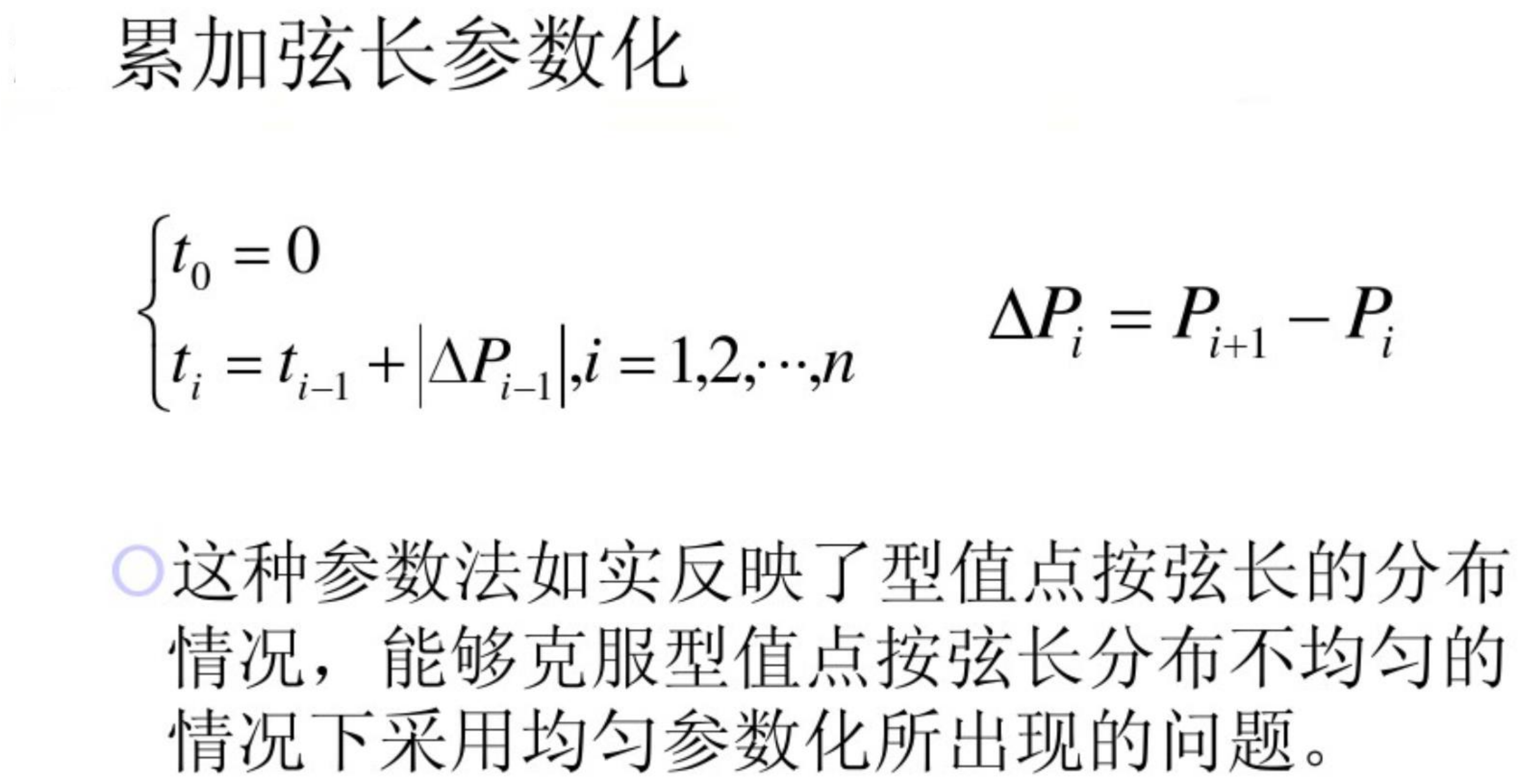
$$t_1 = 0$$

$$t_i = t_{i-1} + \left\| P_i - P_{i-1} \right\|$$

$$i = 2, \cdots, n$$



Least square parametric curve



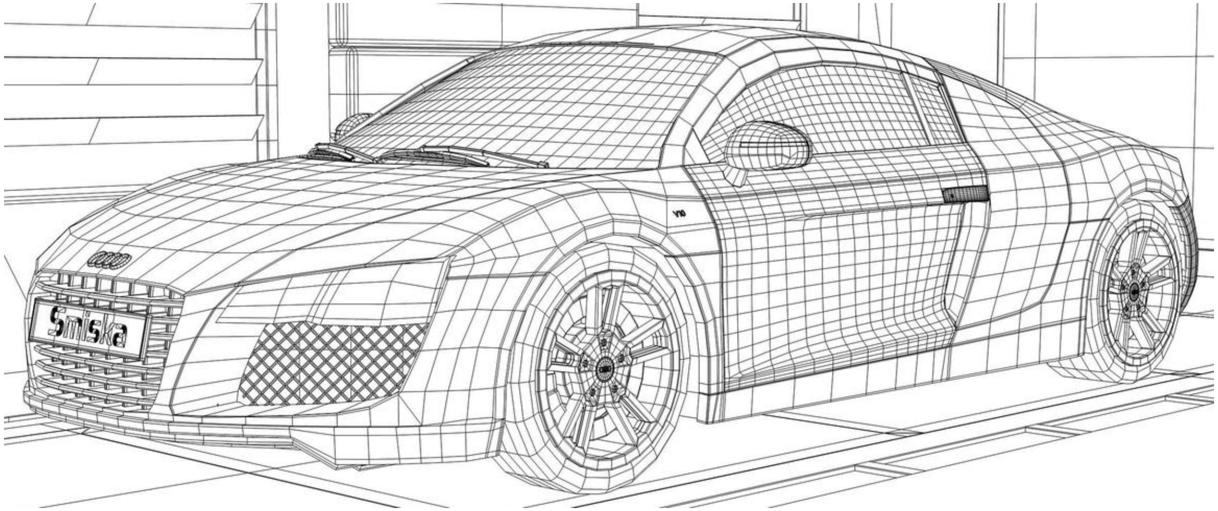
累加弦长参数化

$$\begin{cases} t_0 = 0 \\ t_i = t_{i-1} + \left|\Delta P_{i-1}\right|, i = 1,2,\cdots,n \end{cases} \qquad \Delta P_i = P_{i+1} - P_i$$

○这种参数法如实反映了型值点按弦长的分布情况，能够克服型值点按弦长分布不均匀的情况下采用均匀参数化所出现的问题。

Outline

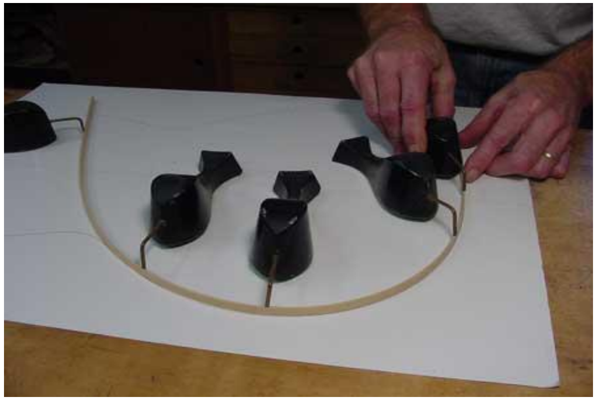
- Interpolation and Approximation
- Curve Modeling
  - Parametric curve
  - Cubic Hermite interpolation
  - Bézier curve
  - B-Spline
- Surface Modeling
  - Bézier surface



Splines - History

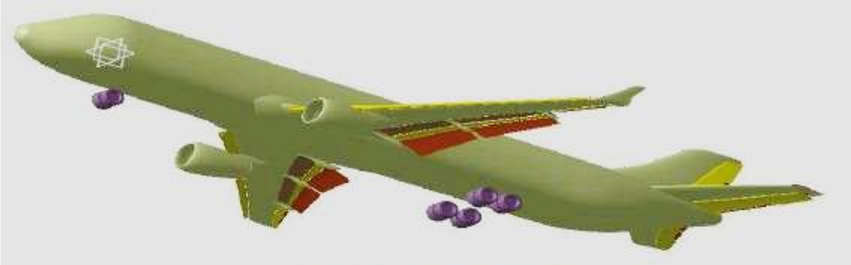
- Draftsman use ‘ducks’ and strips of wood (splines) to draw curves
- Wood splines have second-order continuity
- And pass through the control points

A Duck (weight)



Ducks trace out curve

Spline in industry



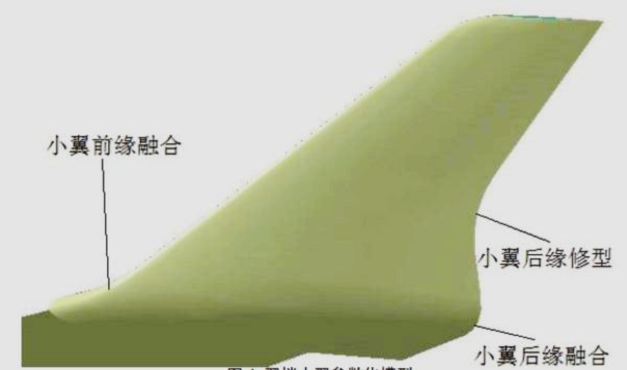


图 4 翼梢小翼参数化模型

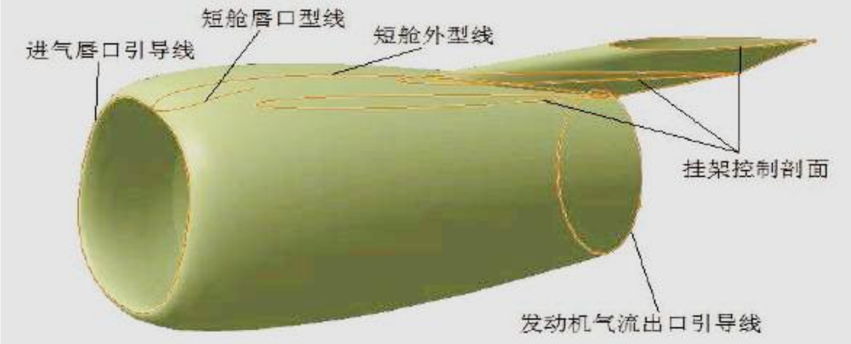


图 5 短舱及挂架参数化模型

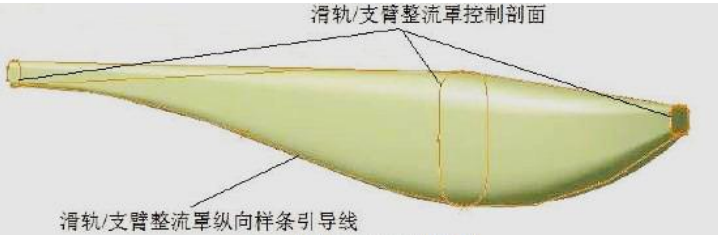


图 6 滑轨整流罩参数化模型

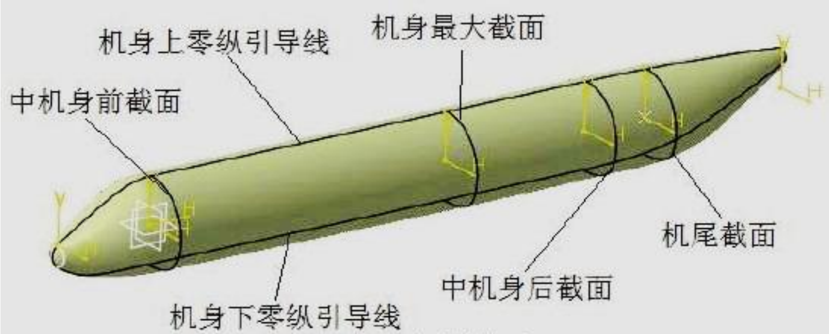
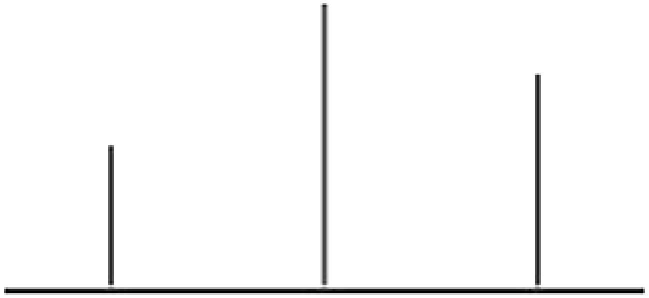


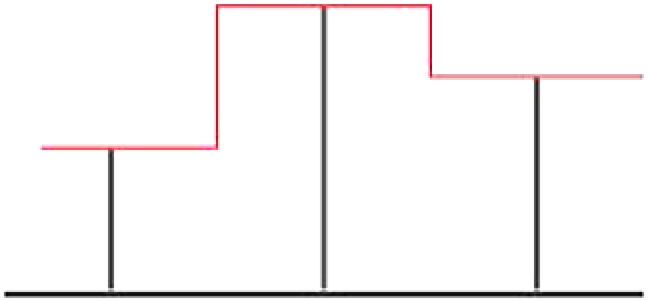
图 3 机身参数模型

Interpolation

- Goal: interpolate values

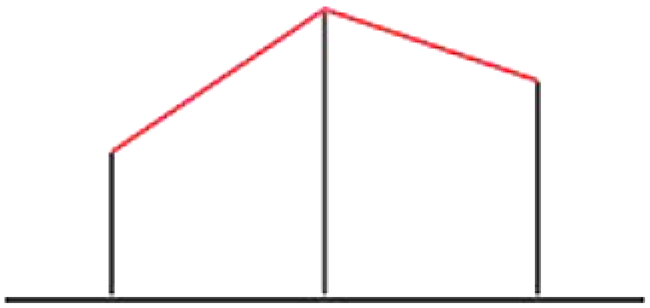


Nearest neighbor interpolation



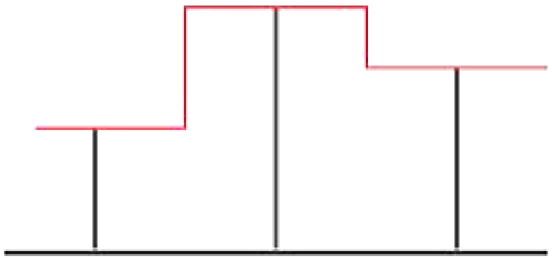
Problem: values not continuous

Linear interpolation

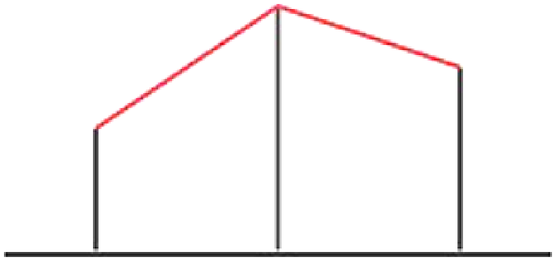


Problem: derivatives not continuous

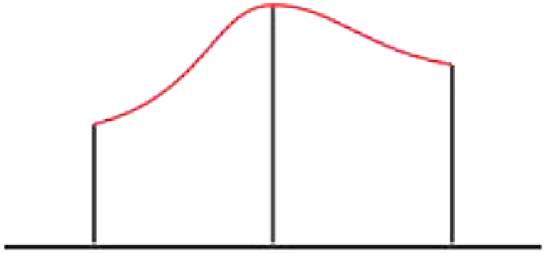
Smooth interpolation?



NN interpolation

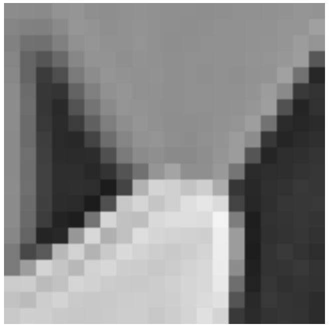


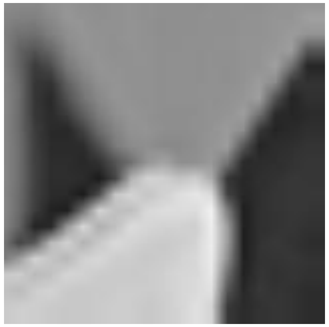
Linear interpolation

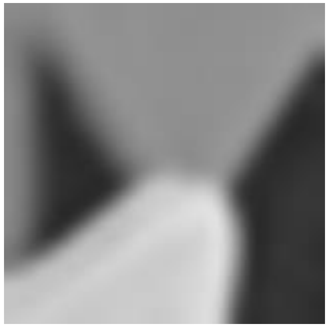


Smooth curve

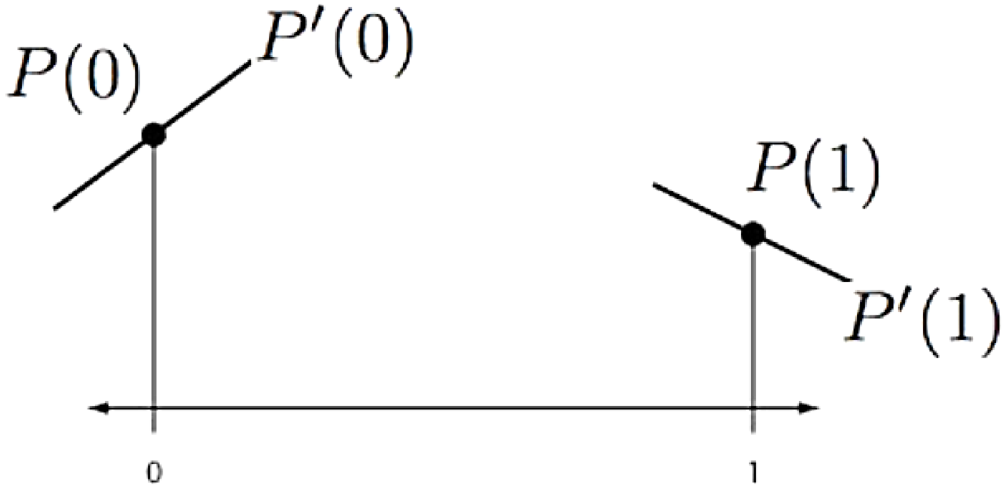
20 x 20 pixel







Cubic Hermite Interpolation



Given: value and derivatives at 2 points

Hermite曲线是通过给定曲线的两个端点的位置矢量P(0)、P(1)以及两个端点处的切线矢量P’(0)、P’(1)来描述曲线。

Cubic Hermite Interpolation

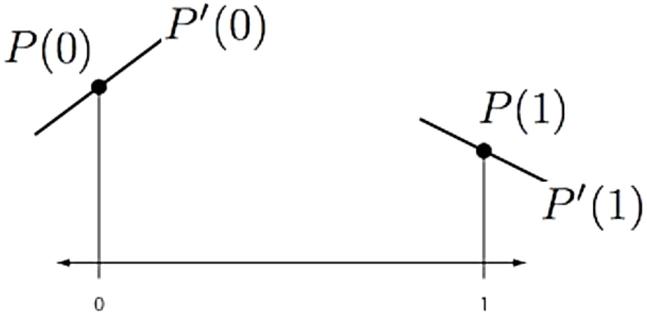
- Assume Cubic polynomial

$$P(t) = a\,t^3 + b\,t^2 + c\,t + d$$

- Solve for coefficients:

$$P(0) = h_0 = d$$

$$P(1) = h_1 = a + b + c + d$$



Cubic Hermite Interpolation

- Cubic polynomial

$$P(t) = a\,t^3 + b\,t^2 + c\,t + d$$

$$P'(t) = 3a\,t^2 + 2b\,t + c$$

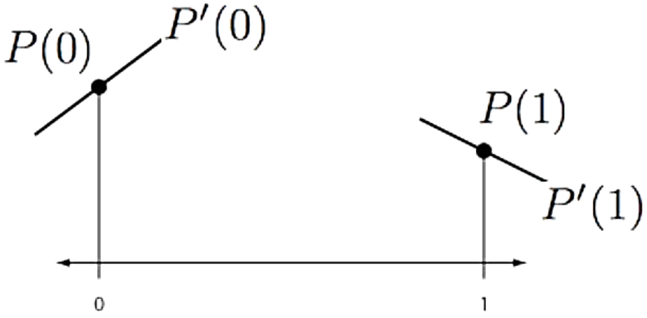
- Solve for coefficients:

$$P(0) = h_0 = d$$

$$P(1) = h_1 = a + b + c + d$$

$$P'(0) = h_2 = c$$

$$P'(1) = h_3 = 3a + 2b + c$$



Matrix Representation of Solution

$$\begin{aligned} h_0 &= d \\ h_1 &= a + b + c + d \\ h_2 &= c \\ h_3 &= 3a + 2b + c \end{aligned}$$

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Solve for a, b, c, d

- Matrix Inverse

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}$$

Matrix Transpose

Transpose $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$

$$\left(\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}\right)^T = \begin{bmatrix} a & b & c & d \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Matrix Representation of Polynomials

- Cubic polynomial

$$P(t) = a\,t^3 + b\,t^2 + c\,t + d$$

$$\boxed{t^0=1}$$

$$\Downarrow$$

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Insert identity matrix

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$\rlap{$\displaystyle\overbrace{\phantom{\begin{bmatrix} a & b & c & d \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}}^{\begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix}}$}\begin{bmatrix} a & b & c & d \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Change Basis

$$\underbrace{\begin{bmatrix} a & b & c & d \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}}_{\begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix}}\underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}\begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}}_{\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}}$$

Hermite Basis Functions

$$\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$H_0(t) = 2t^3 - 3t^2 + 1$$

$$H_1(t) = -2t^3 + 3t^2$$

$$H_2(t) = t^3 - 2t^2 + t$$

$$H_3(t) = t^3 - t^2$$

Hermite Basis Functions

$$\begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}$$

$$P(t) = \sum_{i=0}^{3} h_i \, H_i(t)$$

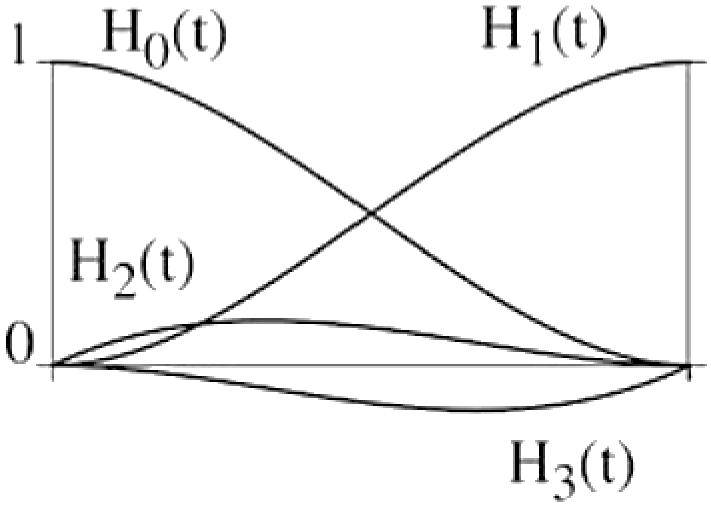
$$P(0) = h_0 = d$$

$$P(1) = h_1 = a + b + c + d$$

$$P'(0) = h_2 = c$$

$$P'(1) = h_3 = 3a + 2b + c$$

Hermite Basis Functions



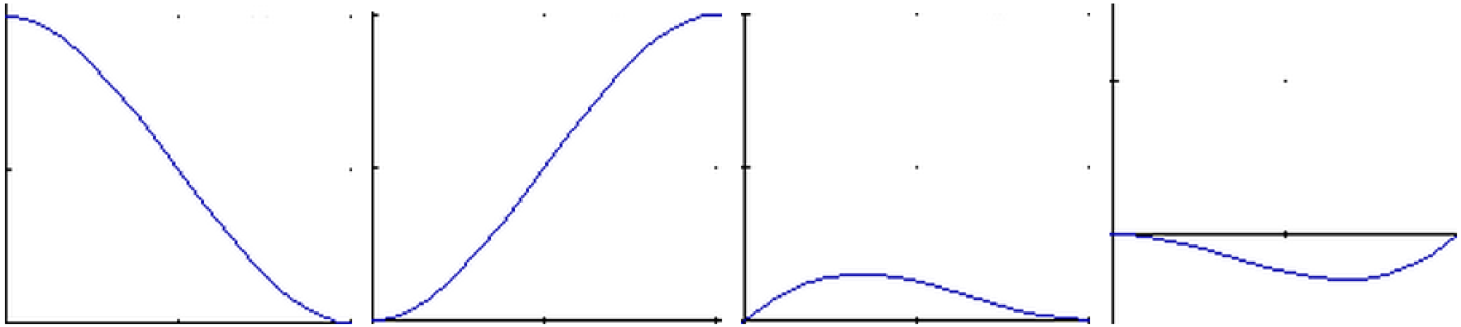
$$H_0(t) = 2t^3 - 3t^2 + 1$$

$$H_1(t) = -2t^3 + 3t^2$$

$$H_2(t) = t^3 - 2t^2 + t$$

$$H_3(t) = t^3 - t^2$$

Below are the 4 graphs of the 4 functions



(all graphs except the 4th have been plotted from 0,0 to 1,1)

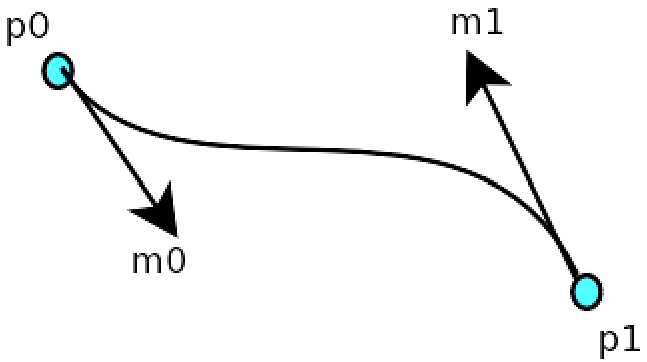
Case

- $\mathbf{P}(t) = (2t^3 - 3t^2 + 1)\ \mathbf{p0}$

  $+ (t^3 - 2t^2 + t)\ \mathbf{m0}$

  $+ (-2t^3 + 3t^2)\ \mathbf{p1}$

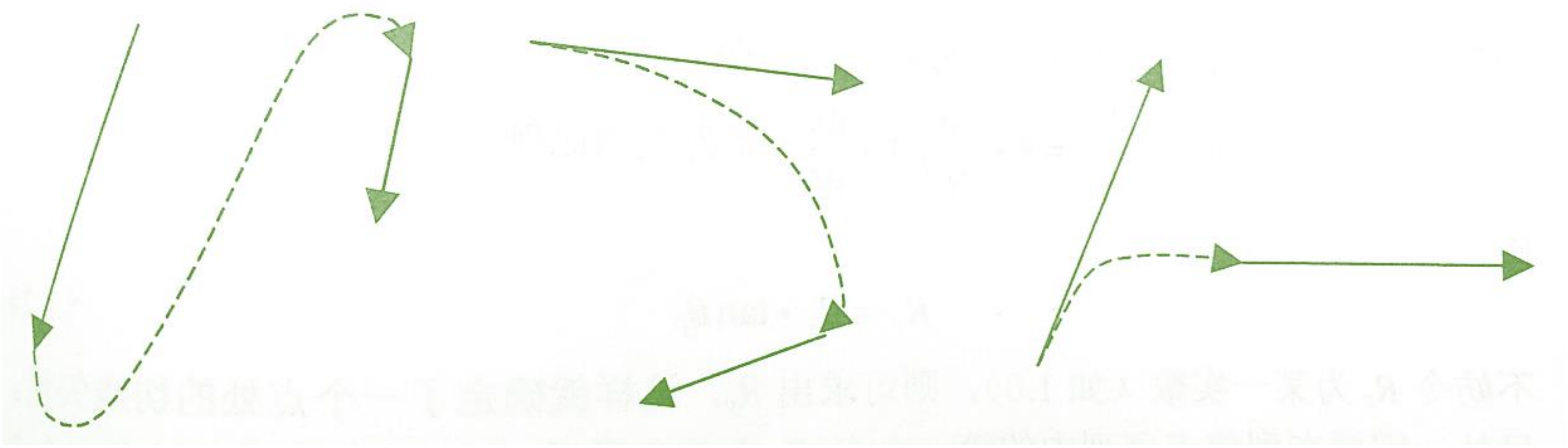
  $+ (t^3 - t^2)\ \mathbf{m1}$



$t \in [0, 1]$

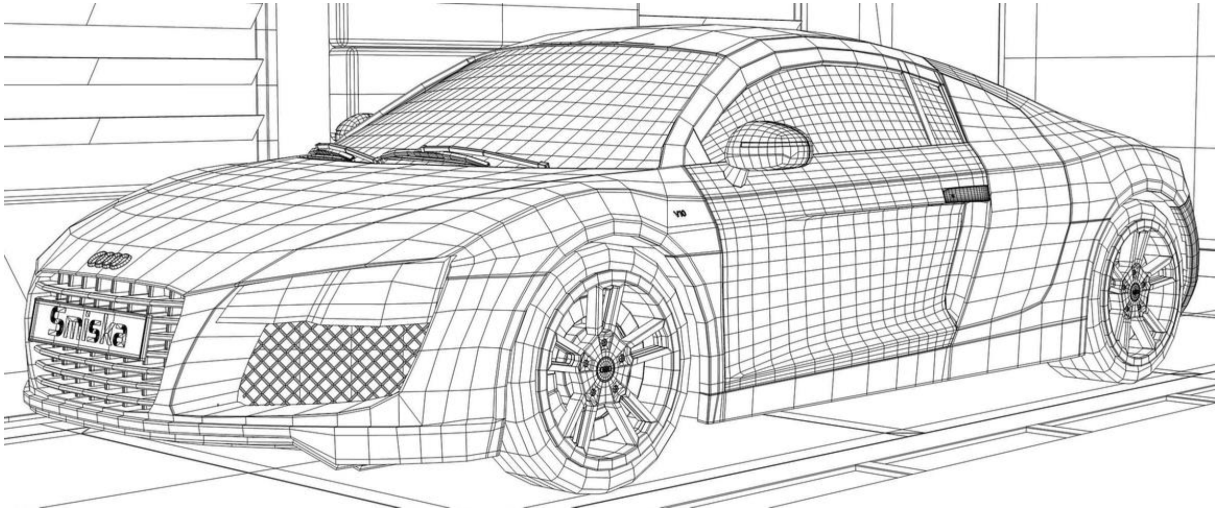
Case

- The derivatives and the shape of Hermite curves



Outline

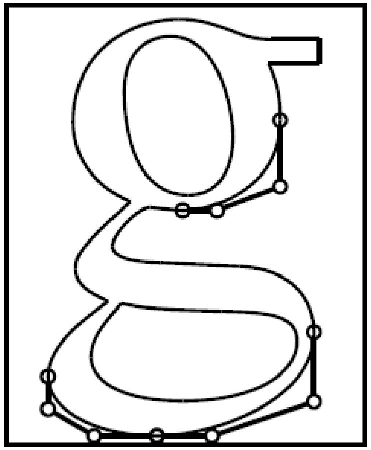
- Interpolation and Approximation
- Curve Modeling
    - Parametric curve
    - Cubic Hermite interpolation
    - Bézier curve
    - B-Spline
- Surface Modeling
    - Bézier surface



Bézier curve

- A Bézier curve is a parametric curve frequently used in computer graphics and related fields.

Pierre Bézier
An engineer at Renault



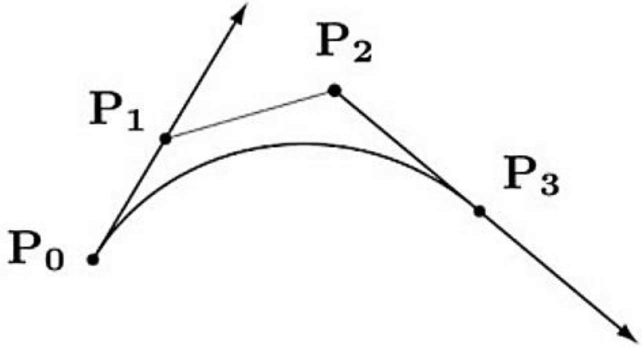
The definition of Bézier curve

- Bézier curve本质上是由**调和函数（Harmonic functions）**根据**控制点（Control points）**插值生成。其参数方程如下：

$$Q(t)=\sum_{i=0}^{n} P_i B_{i,n}(t), \quad t\in[0,1]$$

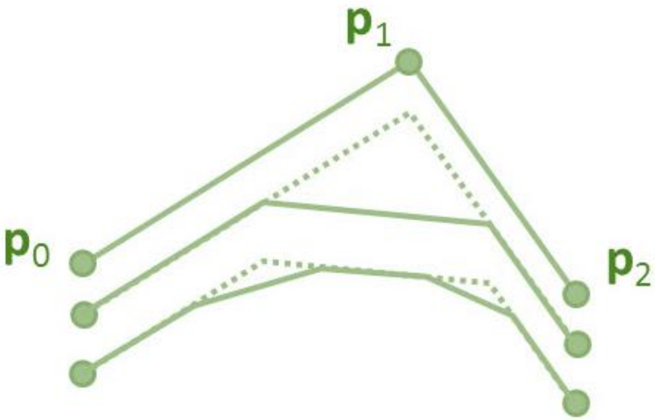
- 上式为n次多项式，具有$n+1$项。其中，$P_i(i=0,1\ldots n)$表示特征多边形的$n+1$个顶点向量；$B_{i,n}(t)$为**伯恩斯坦（Bernstein）基函数**，其多项式表示为：

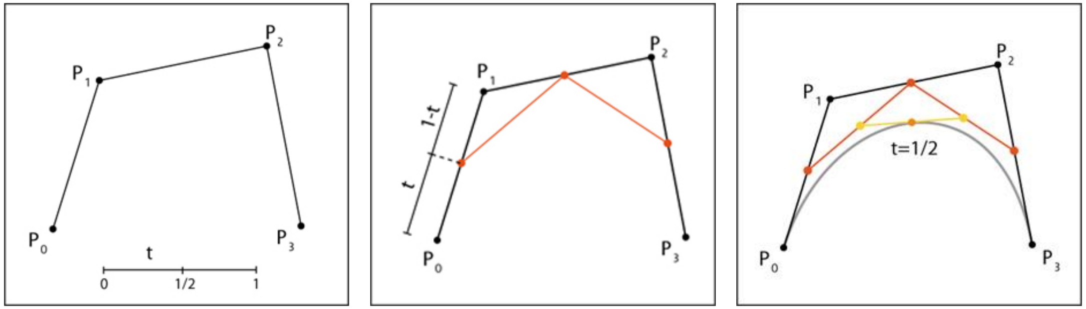
$$B_{i,n}(t)=\frac{n!}{i!(n-i)!}t^i(1-t)^{n-i}, \quad i=0, 1\ldots n$$

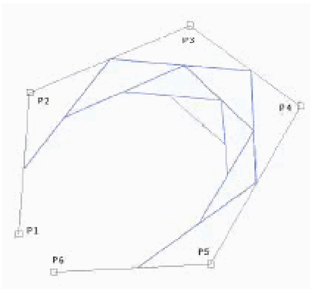


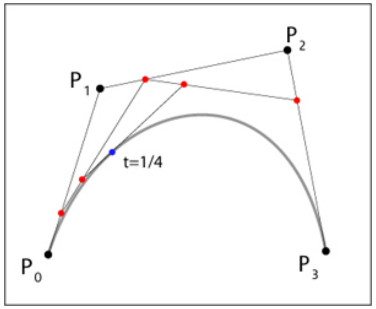
Intuition for Bezier curves

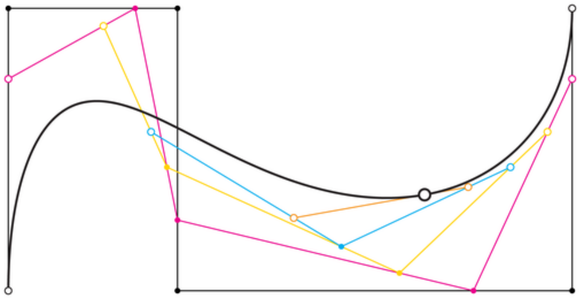
- Keep on cutting corners to make a “smoother” curve
- In the limit, the curve becomes smooth







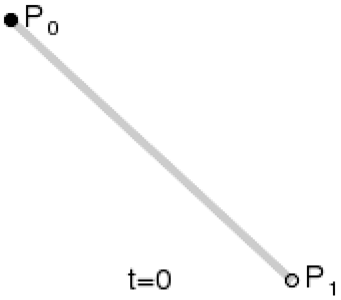




Linear Bézier curve

- Linear polynomial (一次多项式) has two control points, the matrix representation is in the following:

$$
\begin{aligned}
Q(t) &= \sum_{i}^{1} P_i B_{i,1}(t) = P_0 B_{0,1}(t) + P_1 B_{1,1}(t) \\
&= (1-t)P_0 + tP_1 \qquad , \quad t \in [0,1] \\
&= [t \quad 1]\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \end{bmatrix}
\end{aligned}
$$



- Actually, it is a line.

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \text{，} \quad i=0\text{，} \quad 1\ldots n$$

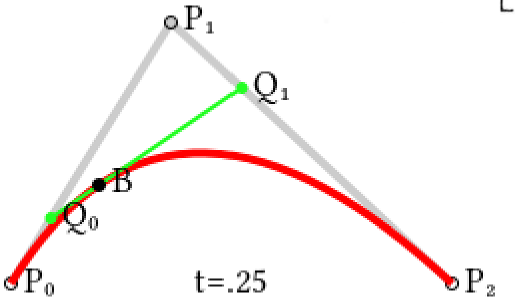
Quadratic Bézier curve

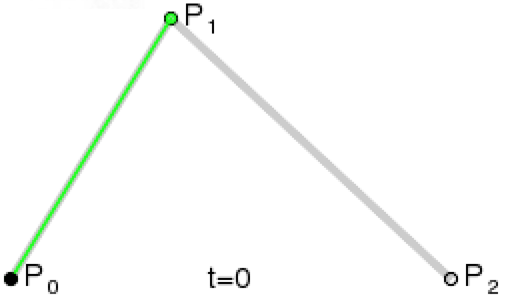
- Quadratic polynomial (二次多项式) has 3 control points, the math formula is as follows:

$$
\begin{aligned}
Q(t) &= \sum_{i}^{2} P_i B_{i,2}(t) = P_0 B_{0,2}(t) + P_1 B_{1,2}(t) + P_2 B_{2,2}(t) \\
&= (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2 \qquad , \quad t \in [0,1] \\
&= (P_2 - 2P_1 + P_0)t^2 + 2(P_1 - P_0)t + P_0
\end{aligned}
$$

- Quadric Bézier curve is parabola，It’s matrix representation:

$$
Q(t) = [t^2 \quad t \quad 1]\begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix}, \quad t \in [0,1]
$$





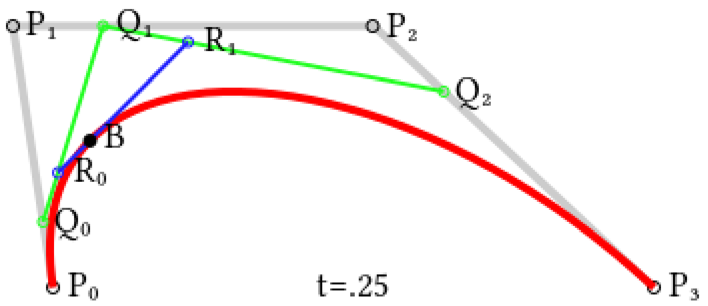
Cubic Bézier curve

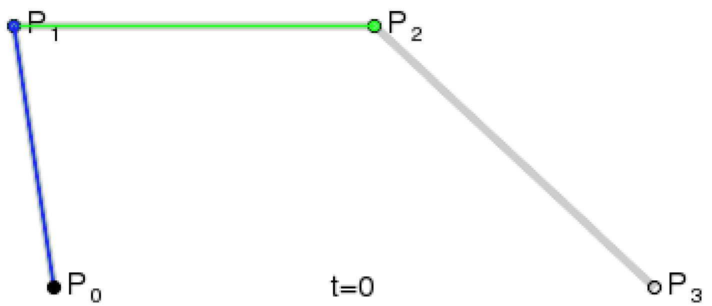
- Cubic polynomial (二次多项式) has 4 control points, the math formula is as follows:

$$Q(t)=\sum_{i}^{3}P_iB_{i,3}(t)=P_0B_{0,3}(t)+P_1B_{1,3}(t)+P_2B_{2,3}(t)+P_3B_{3,3}(t)\ ,\quad t\in[0,1]$$
$$=(1-t)^3P_0+3t(1-t)^2P_1+3t^2(1-t)P_2+t^3P_3$$

- The matrix representation:

$$Q(t)=[t^3\quad t^2\quad t\quad 1]\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 3 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}P_0\\ P_1\\ P_2\\ P_3\end{bmatrix},\quad t\in[0,1]$$

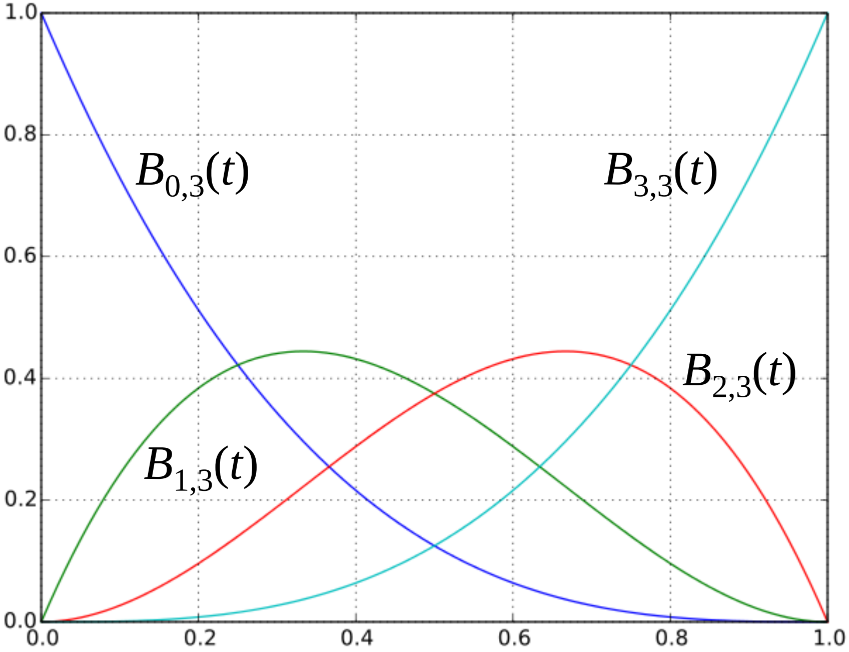




Bernstein Basis Functions

- 根据Bernstein多项式构成了三次Bézier曲线的一组基，或称为三次Bézier曲线的调和函数，即：

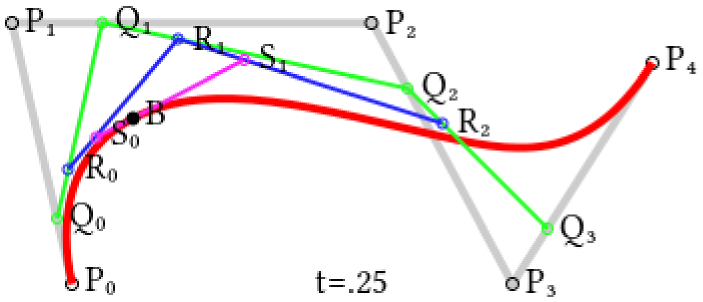
$$\begin{cases} B_{0,3}(t) = (1-t)^3 \\ B_{1,3}(t) = 3t(1-t)^2 \\ B_{2,3}(t) = 3t^2(1-t) \\ B_{3,3}(t) = t^3 \end{cases}$$

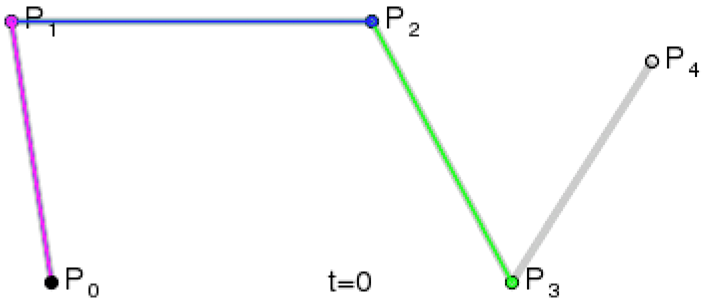


The basis functions of cubic Bézier curve on the range t in [0,1]

High-order Bézier curve

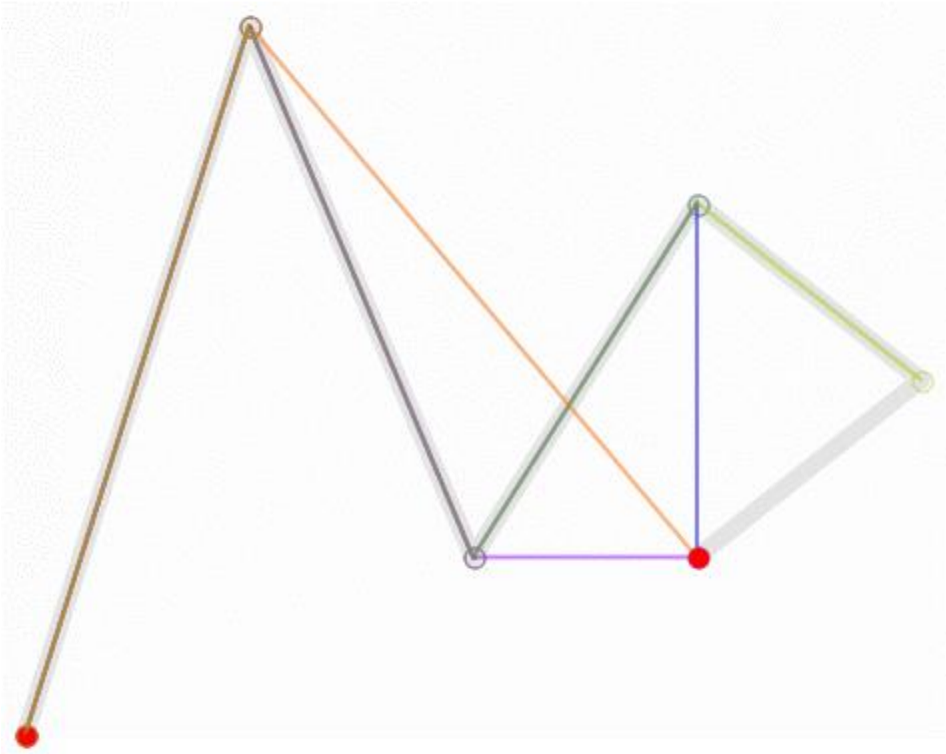
- For fourth-order curves one can construct intermediate points **Q**$_0$, **Q**$_1$, **Q**$_2$ & **Q**$_3$ that describe linear Bézier curves, points **R**$_0$, **R**$_1$ & **R**$_2$ that describe quadratic Bézier curves, and points **S**$_0$ & **S**$_1$ that describe cubic Bézier curves:





High-order Bézier curve

- For fifth-order curves, one can construct similar intermediate points.



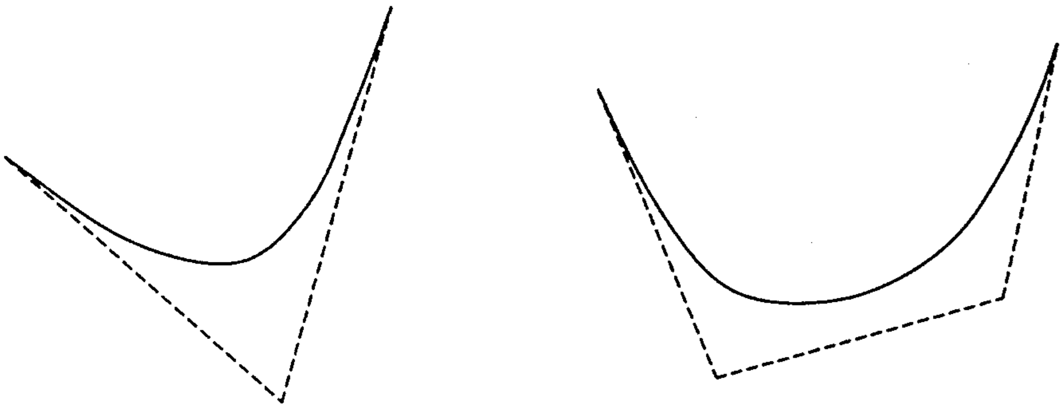
The properties of Bézier curve

- 1. 端点性质：当$t = 0$和$t = 1$时，有：

$$Q(0) = \sum_{i=0}^{n} P_i B_{i,n}(0) = P_0 B_{0,n}(0) + P_1 B_{1,n}(0) + \cdots + P_n B_{n,n}(0) = P_0$$

$$Q(1) = \sum_{i=0}^{n} P_i B_{i,n}(1) = P_0 B_{0,n}(1) + P_1 B_{1,n}(1) + \cdots + P_n B_{n,n}(1) = P_n$$

- 这说明，Bézier曲线通过特征多边形的起点和终点。



The properties of Bézier curve

- 2. 对称性：由于$B_{i,n}(t) = B_{n-i,n}(1-t)$，如果将控制点的顺序颠倒过来，记$P_i^* = P_{n-i}$,则根据Bézier曲线的定义可推出：

$$
\begin{aligned}
Q^*(t) &= \sum_{i=0}^{n} P_i^* B_{i,n}(t) = \sum_{i=0}^{n} P_{n-i} B_{i,n}(t) = \sum_{i=n}^{0} P_k B_{n-k,n}(t) \\
&= \sum_{i=n}^{0} P_k B_{k,n}(1-t) \\
&= Q(1-t)
\end{aligned}
$$

- 这说明，只要保持特征多边形的顶点位置不变，但顺序颠倒，所得的新的Bézier曲线形状不变，只是参数变化的方向相反。

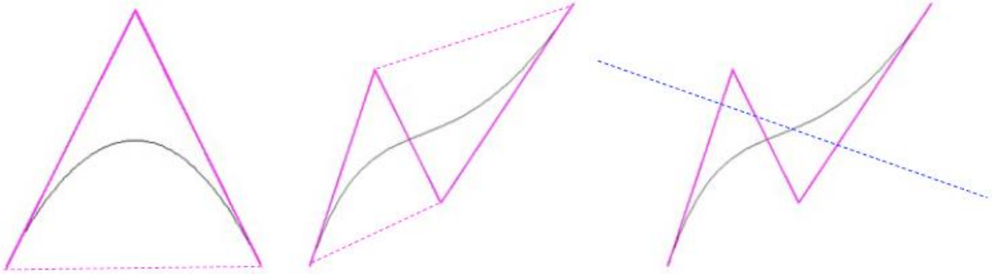
The properties of Bézier curve

- 凸包性：由于当$t \in [0,1]$时，Bernstein多项式之和为：

$$\sum_{i=0}^{n} B_{i,n}(t) = \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} t^{i} (1-t)^{n-i}$$

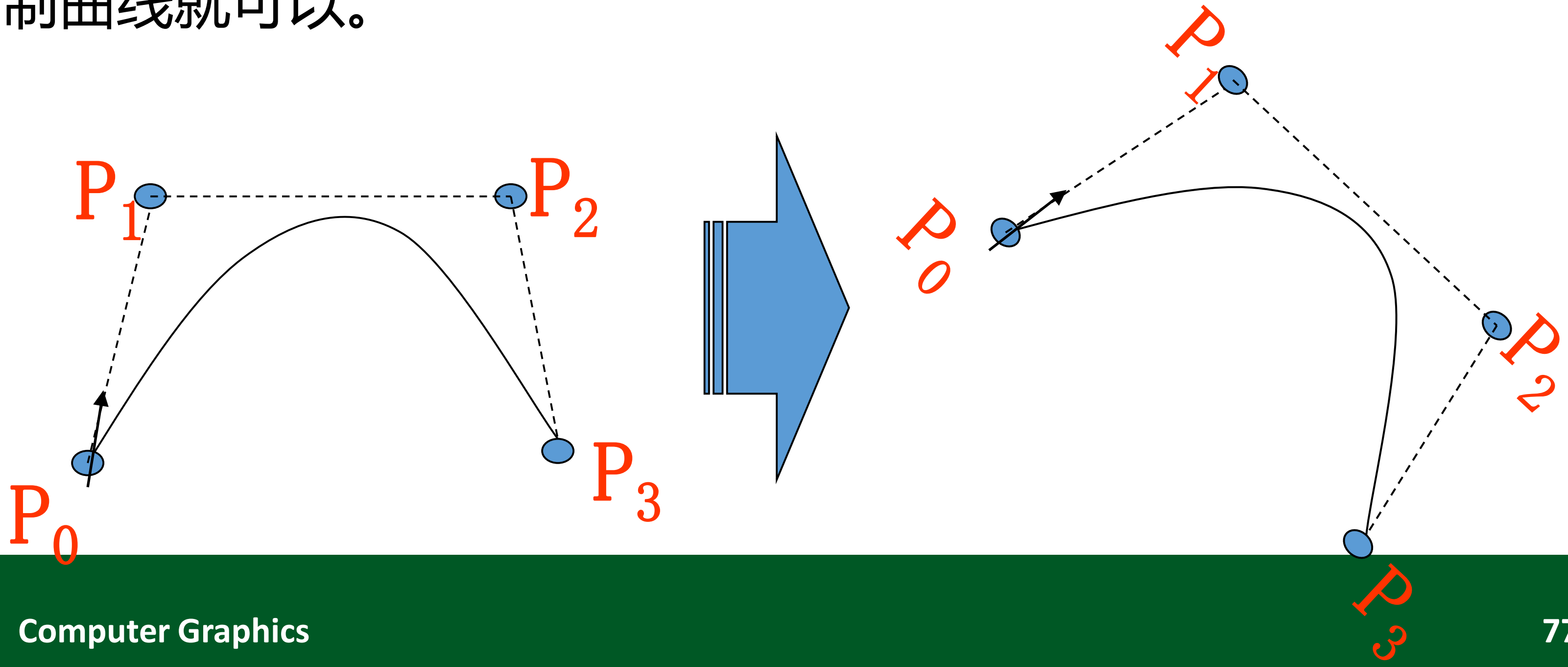
$$= [(1-t)+t] \equiv 1$$

- 且有 $B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^{i} (1-t)^{n-i} \geqslant 0$
- 则说明$B_{i,n}(t)$构成了Bézier曲线的一组权函数，所以Bézier曲线一定落在其控制多边形的凸包之中。

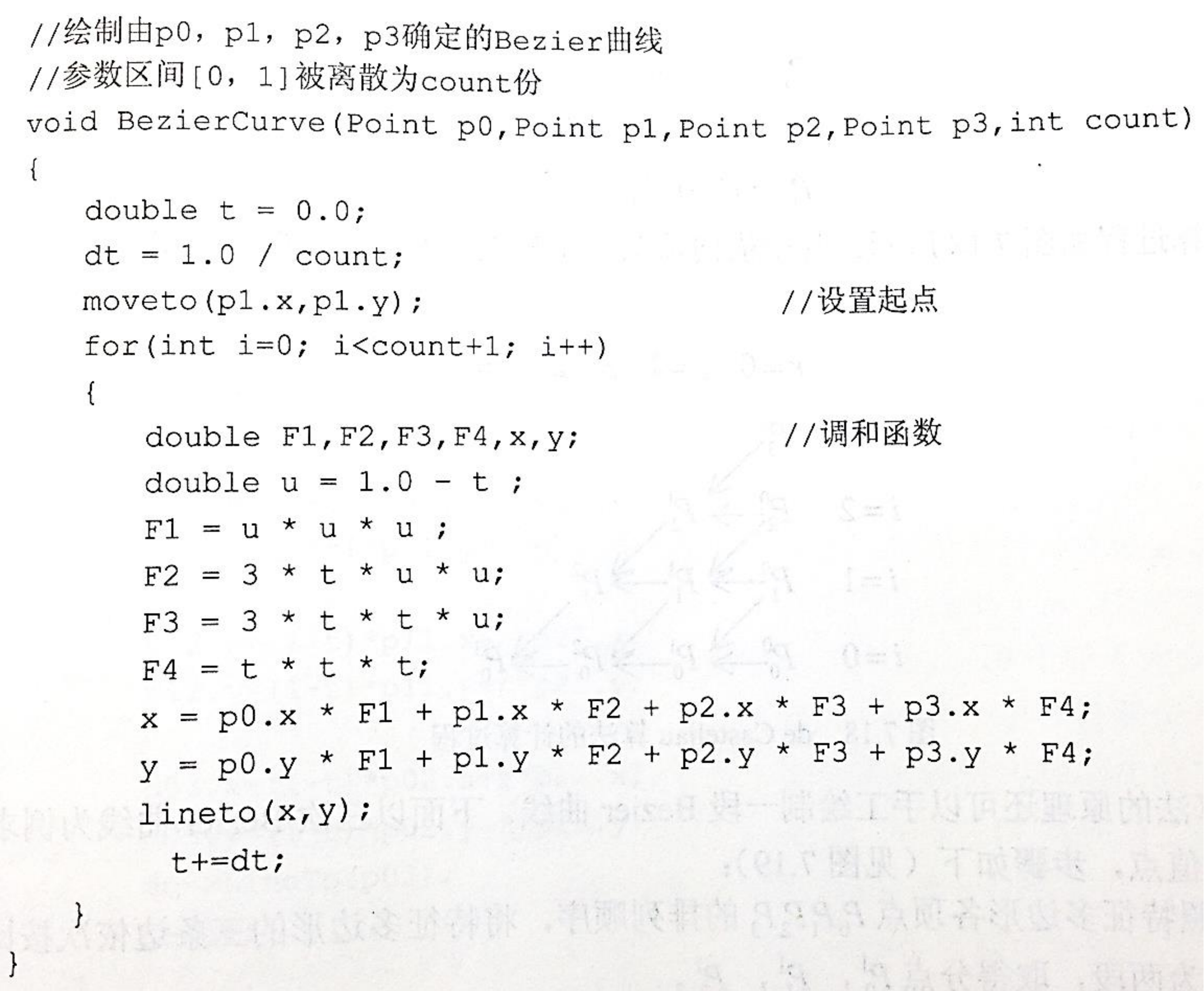


The properties of Bézier curve

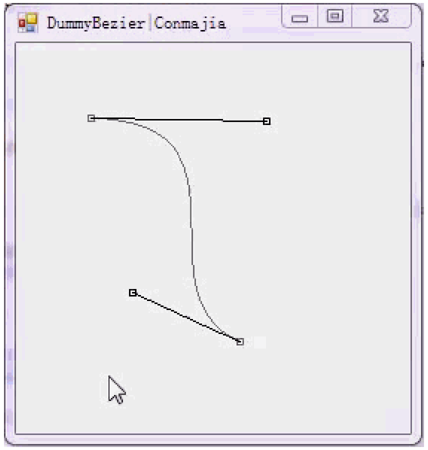
- 几何不变形（Geometric Invariant）：指Bézier曲线的形状不随坐标变换而变化的特性。 Bézier曲线的形状只与各控制顶点的相对位置有关。
- 因此，在对Bézier曲线进行几何变换时，不需要对曲线上的所有点都进行处理，只需要先对控制顶点进行几何变换，然后重新绘制曲线就可以。



Implementation – Cubic Bézier curve

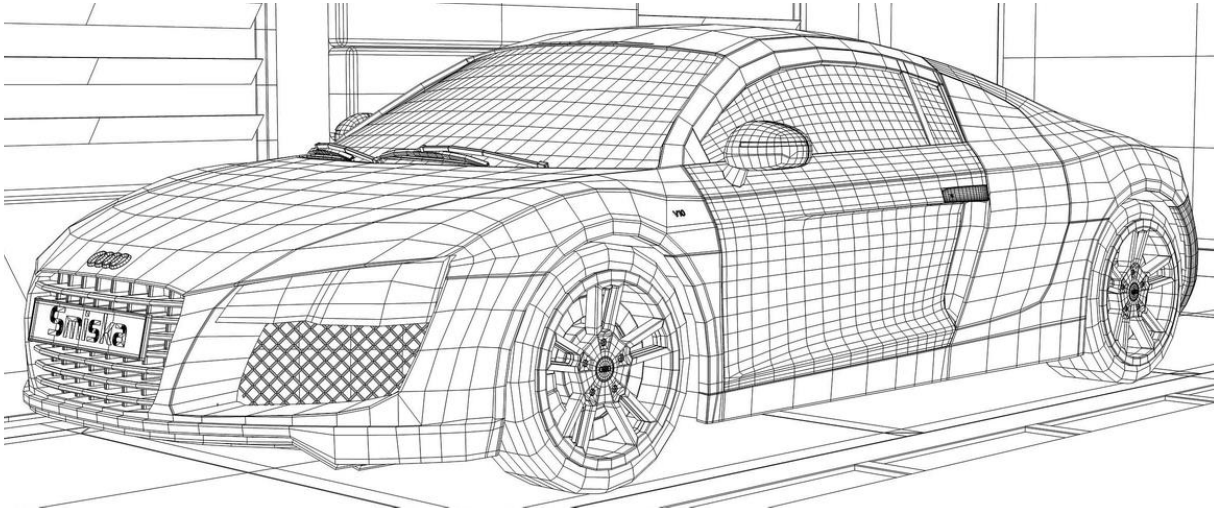


```
//绘制由p0，p1，p2，p3确定的Bezier曲线
//参数区间[0，1]被离散为count份
void BezierCurve(Point p0,Point p1,Point p2,Point p3,int count)
{
    double t = 0.0;
    dt = 1.0 / count;
    moveto(p1.x,p1.y);                        //设置起点
    for(int i=0; i<count+1; i++)
    {
        double F1,F2,F3,F4,x,y;               //调和函数
        double u = 1.0 - t ;
        F1 = u * u * u ;
        F2 = 3 * t * u * u;
        F3 = 3 * t * t * u;
        F4 = t * t * t;
        x = p0.x * F1 + p1.x * F2 + p2.x * F3 + p3.x * F4;
        y = p0.y * F1 + p1.y * F2 + p2.y * F3 + p3.y * F4;
        lineto(x,y);
          t+=dt;
    }
}
```



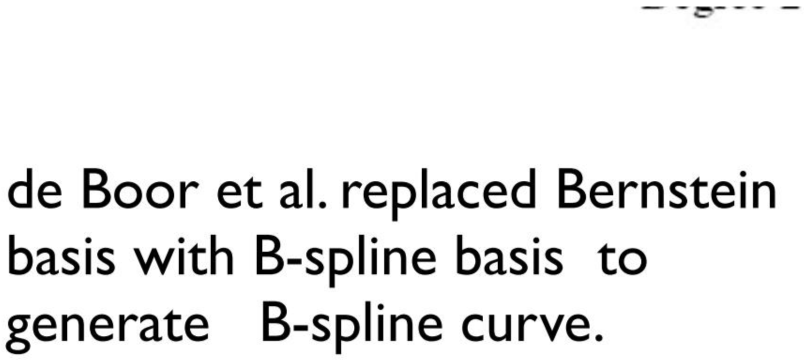
Outline

- Interpolation and Approximation
- Curve Modeling
  - Parametric curve
  - Cubic Hermite interpolation
  - Bézier curve
  - B-Spline
- **Surface Modeling**
  - Bézier surface



Why to introduce B-Spline?

- Bezier curve has many advantages, but they have two main shortcomings:
  - The number of control points determines the degree of the curve. many control points means high degree.
  - It's global. A control point influences the whole curve.
- B-spline curves do not suffer these drawbacks.

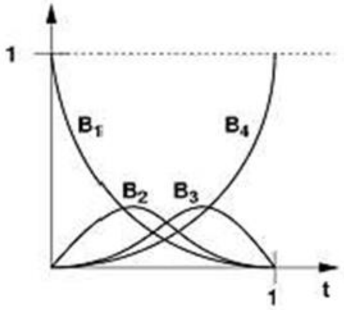


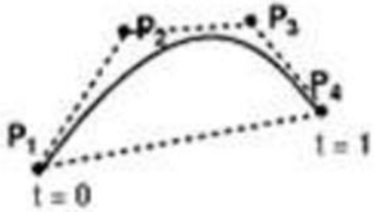
de Boor et al. replaced Bernstein basis with B-spline basis to generate B-spline curve.

B-Spline Curve

- 我们先来实际体会一下B样条曲线和Bezier曲线的差别，看看下面例子：

Bézier

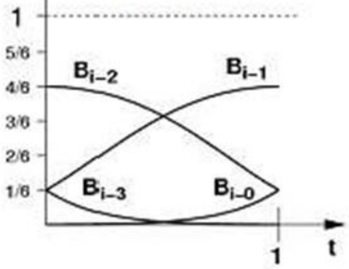


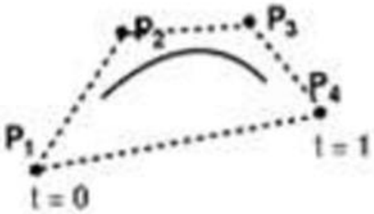


3次曲线为例，基函数不同
3次曲线都需要至少4个控制点

B样条也是逼近曲线，不一定过控制点，甚至不过起控制点和终控制点

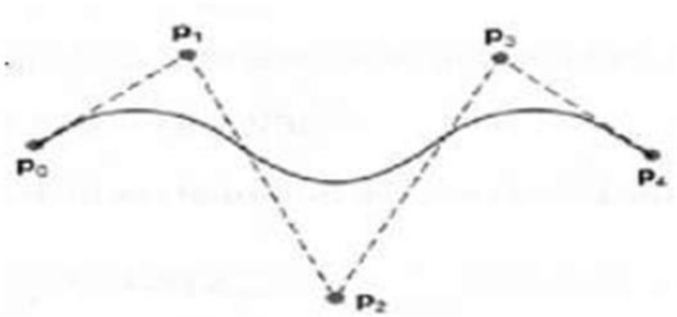
BSpline

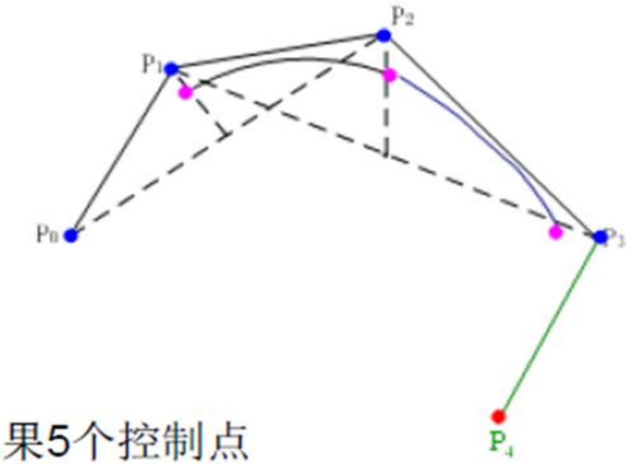




B-Spline Curve

- **我们先来实际体会一下B样条曲线和Bezier曲线的差别，看看下面例子：**





Bezier曲线如果5个控制点
那么只能是4次曲线
条件任何一个控制点，
会影响整个4次曲线

B样条曲线如果5个控制点

可以使用3次曲线，也可以使用4次曲线来构造整个曲线

使用4次曲线那么就是1段曲线

使用3次曲线那么就构造2段曲线，并且这2段曲线可以自然拼接起来，调节P4点位置只会影响第二段曲线

Definition of B-splines

- B-spline curve is piecewise polynomial curve
- Given knot vector(节点向量) :

u={u0, u1, …, ui, …, un+k+1 }

A B-spline of degree k (order k+1) with (n+1) control points is defined as

$$\mathbf{R}(u)=\sum_{i=0}^{n}\mathbf{R}_i N_{i,k}(u) \qquad u \in \left[u_0,\ u_{n+k+1}\right]$$

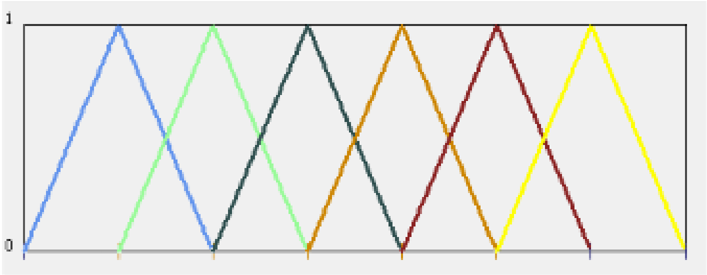
K次或k+1阶B样条

Definition of B-splines

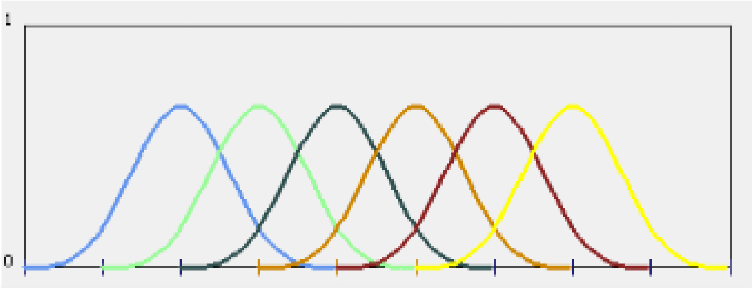
- $\mathrm{R}_{i:}$ control points， $\{\mathrm{R}_i\}_{i=0,1,\ldots,n}$: control polygon
- $N_{i,k}(u)$ are basis of B-spline：

$$\begin{cases} N_{i,0} = \begin{cases} 1 & \text{当} u_i \le u < u_{i+1} \\ 0 & \text{其它} \end{cases} \\ N_{i,k}(u) = \dfrac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \dfrac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \\ \text{定义} \dfrac{0}{0} = 0 \end{cases}$$

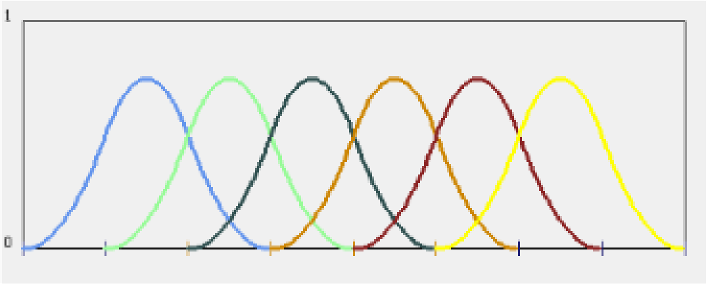
B-splines Basis



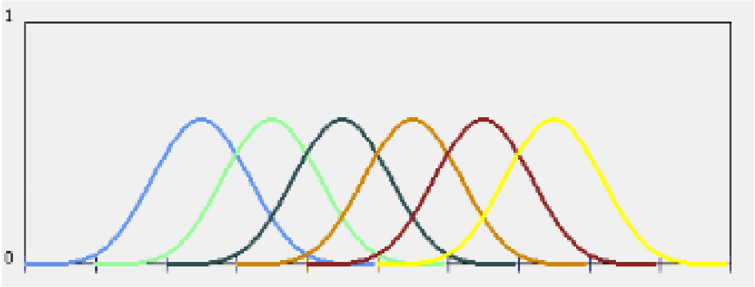
一次B 样条曲线的基函数



三次B 样条曲线的基函数

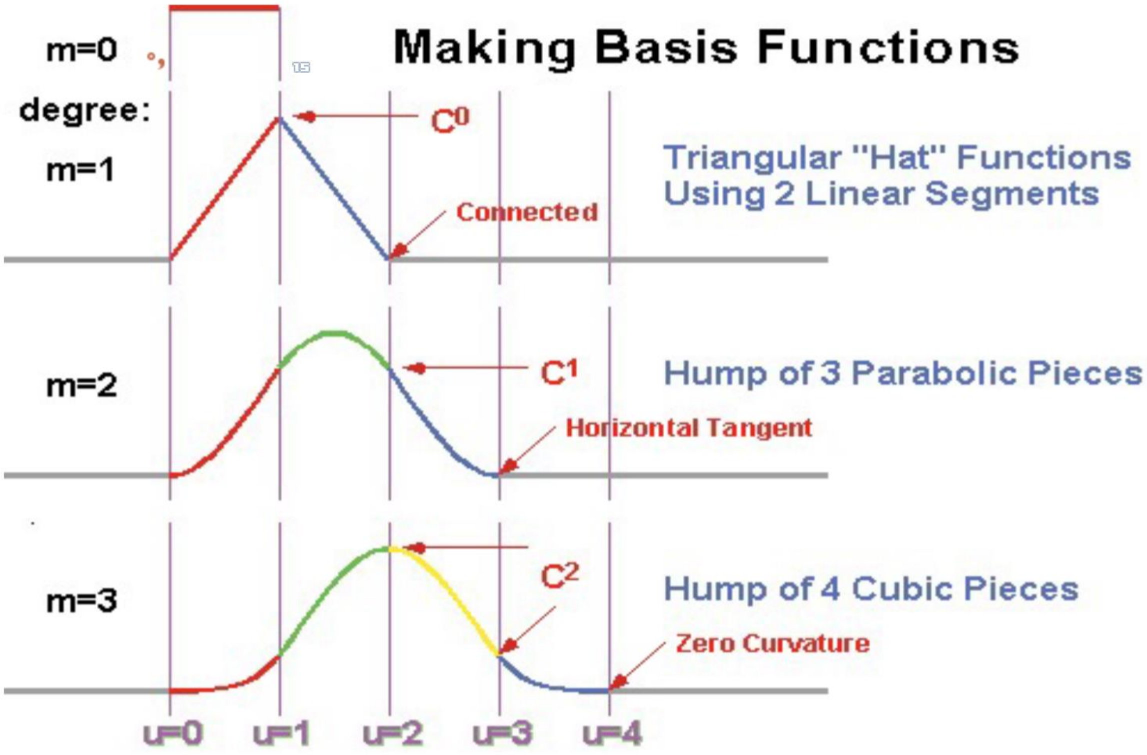


二次B 样条曲线的基函数



四次B 样条曲线的基函数

B-splines Basis



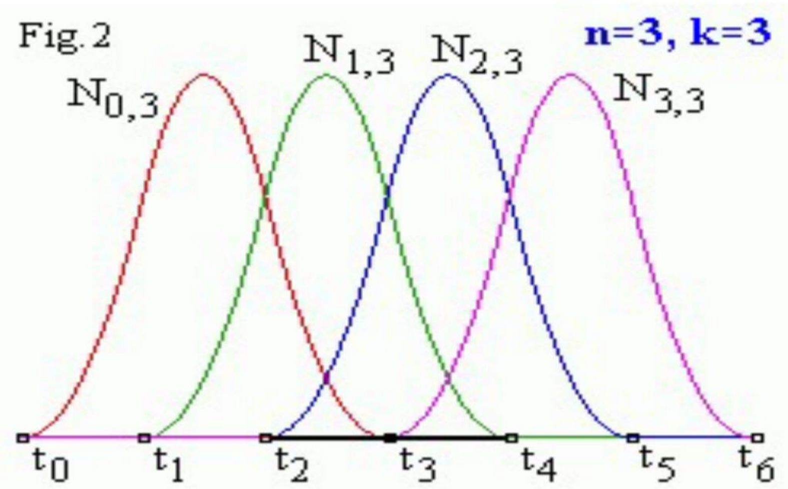
$$U = \{u_i\}_{i=-\infty}^{\infty}$$

$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & else \end{cases}$$

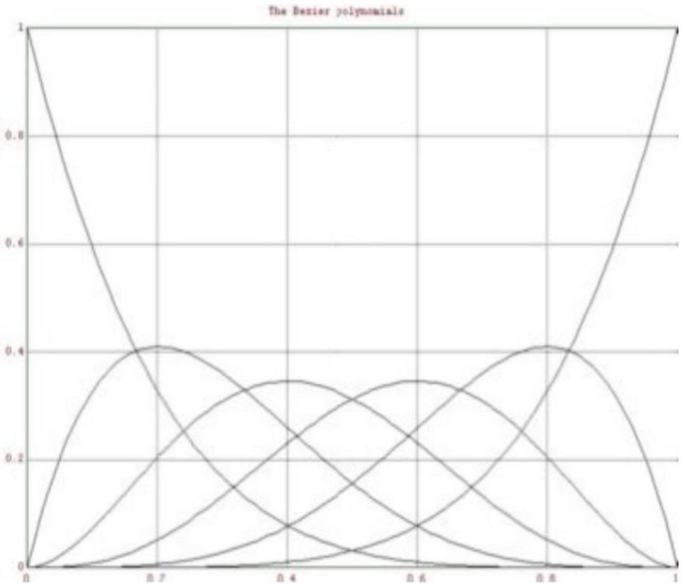
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u),$$

$$\frac{0}{0} = 0$$

B-splines Basis Vs. Bernstein



$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t),\ i = 0,1,\ldots,n.$$



$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & else \end{cases}$$

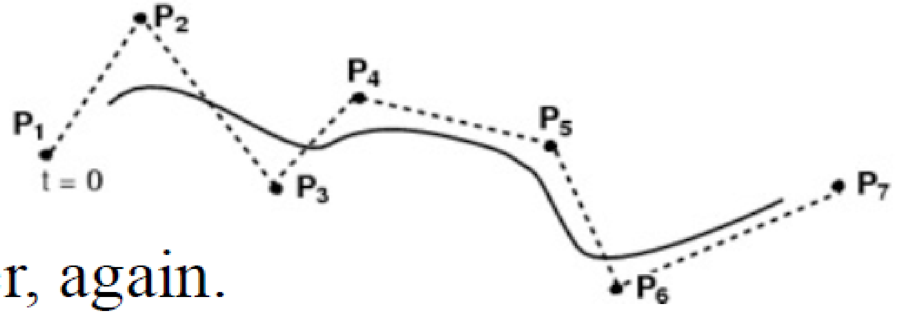
$$U = \{u_i\}_{i=-\infty}^{\infty}$$

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u),$$

$$\frac{0}{0} = 0$$

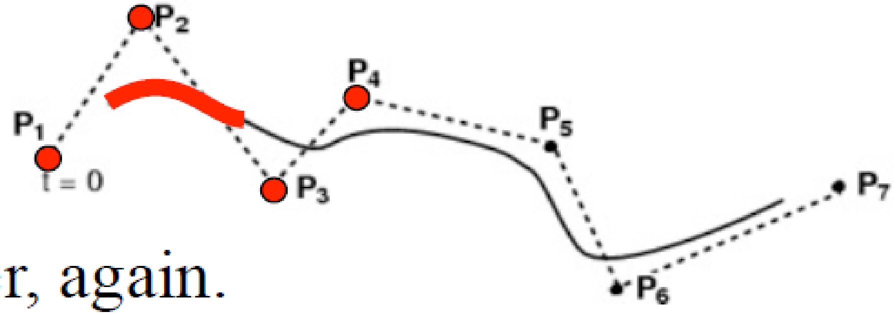
Cubic B-Splines

- $\geq$ 4 control points
- Locally cubic
  - – Cubics chained together, again.



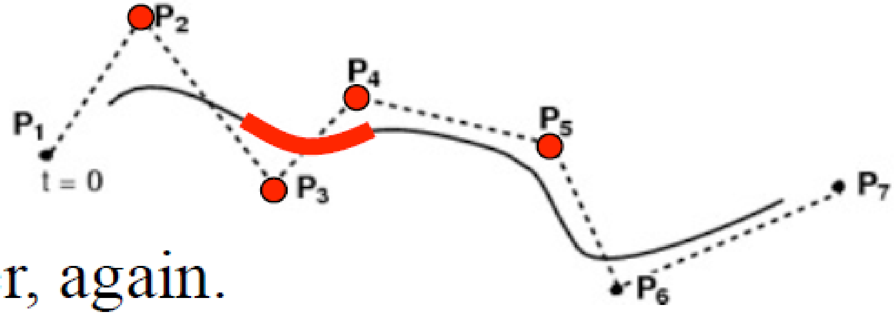
Cubic B-Splines

- $\geq$ 4 control points
- Locally cubic
  - – Cubics chained together, again.



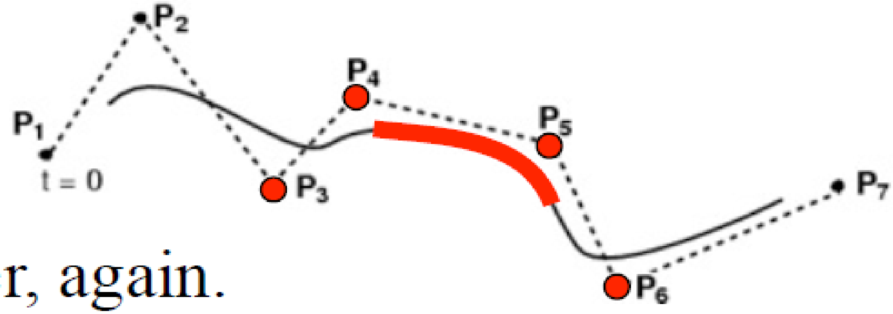
Cubic B-Splines

- ≥ 4 control points
- Locally cubic
  - – Cubics chained together, again.



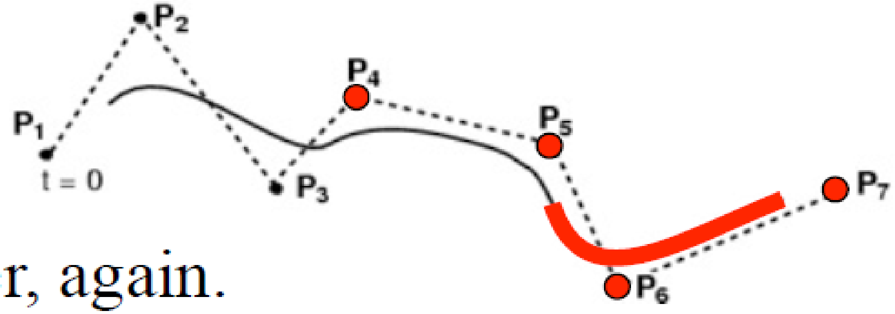
Cubic B-Splines

- ≥ 4 control points
- Locally cubic
  - – Cubics chained together, again.

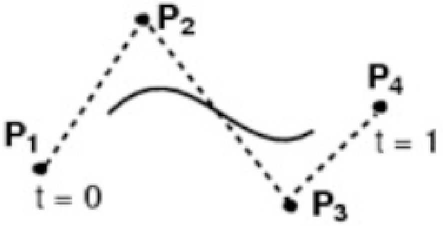


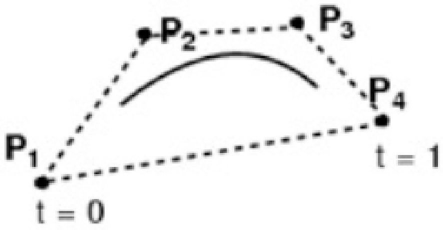
Cubic B-Splines

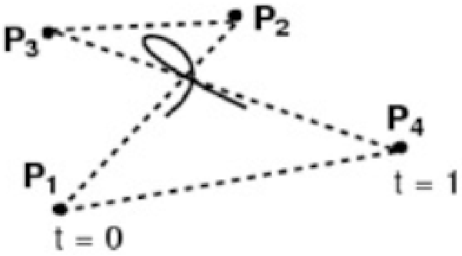
- ≥ 4 control points
- Locally cubic
  - Cubics chained together, again.



- Curve is not constrained to pass through any control points



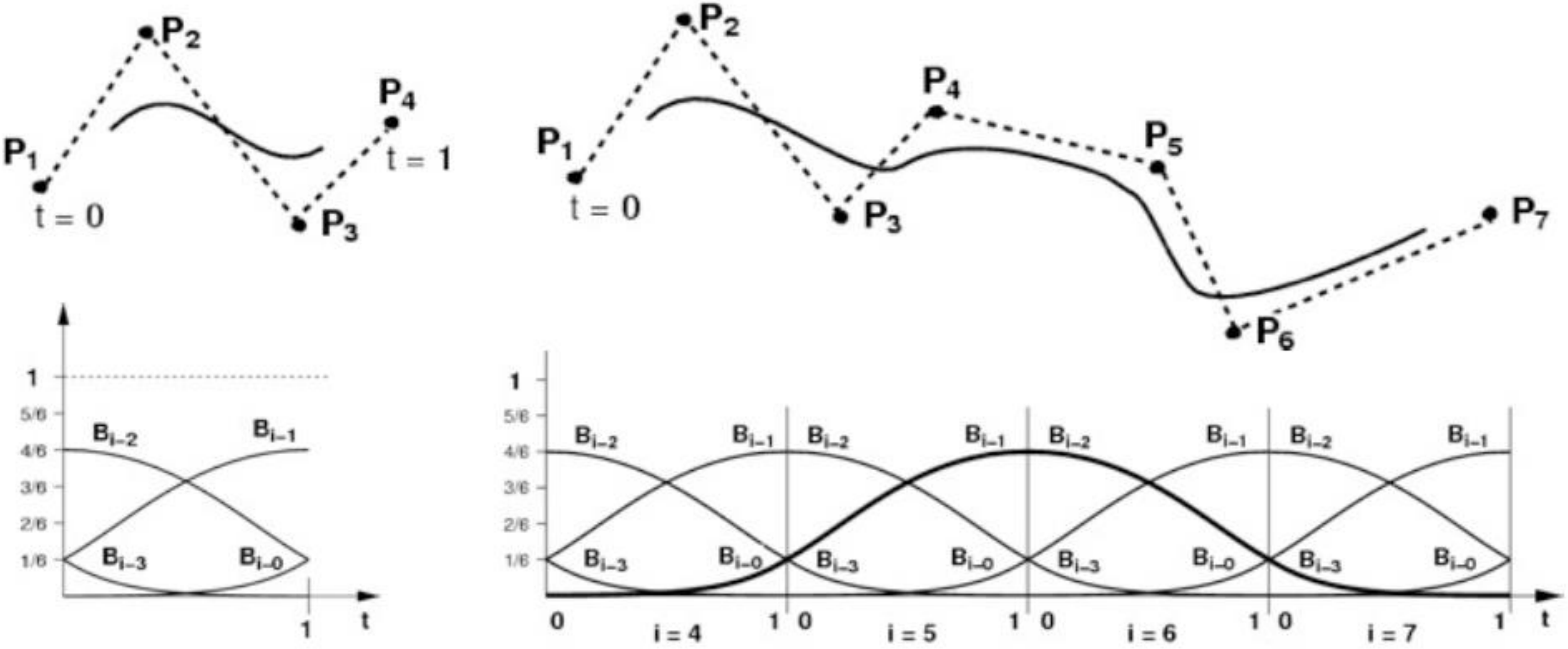




A BSpline curve is also bounded by the convex hull of its control points.

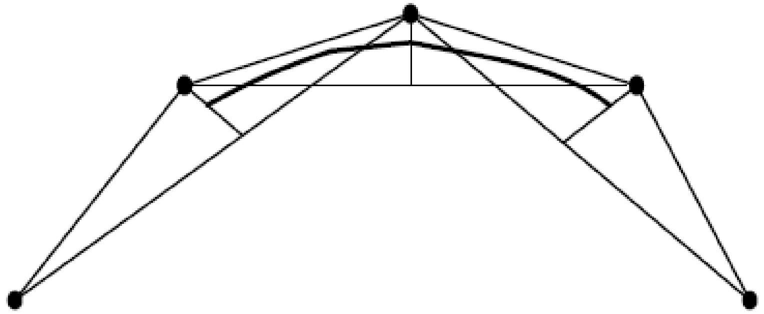
Cubic B-Splines

- Local control (windowing)
- Automatically C^2, and no need to match tangents!



Kinds of B-Spline

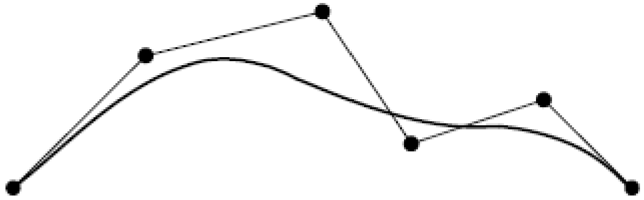
- (1) Uniform B-Spline
  - The knots are uniformed distributed, like 0,1,2,3,4,5,6,7
  - This kind of knot vector defines uniform B-Spline basis function



uniform B-Spline of Degree 3

Kinds of B-Spline

- (2) Quasi-Uniform B-Spline
  - Different from uniform B-Spline, it has:
    - the start-knot and end-knot have repetitiveness(重复度) of k
    - Uniform B-Spline does not retain the “end point” property of Bezier Curve, which means the start point and end point of uniform B-Spline are no-longer the same as the start point and end point of the control points. However, quasi-Uniform B-Spline retains this “end point” property

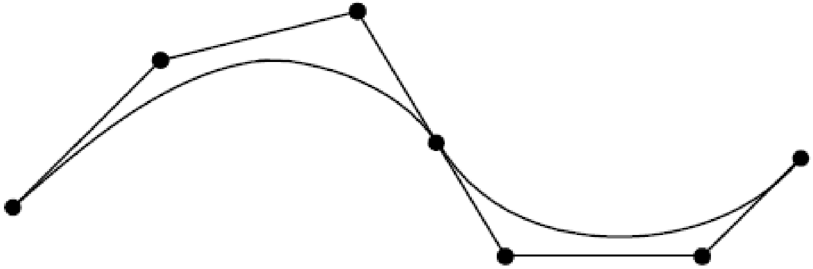


Quasi-uniform B-Spline curve of degree 3

Kinds of B-Spline

– (3) Piecewise Bezier Curve

- the start-knot and end-knot have repetitiveness(重复度) of k
- all other knots have repetitiveness of k-1
- Then each curve segment will be Bezier curves



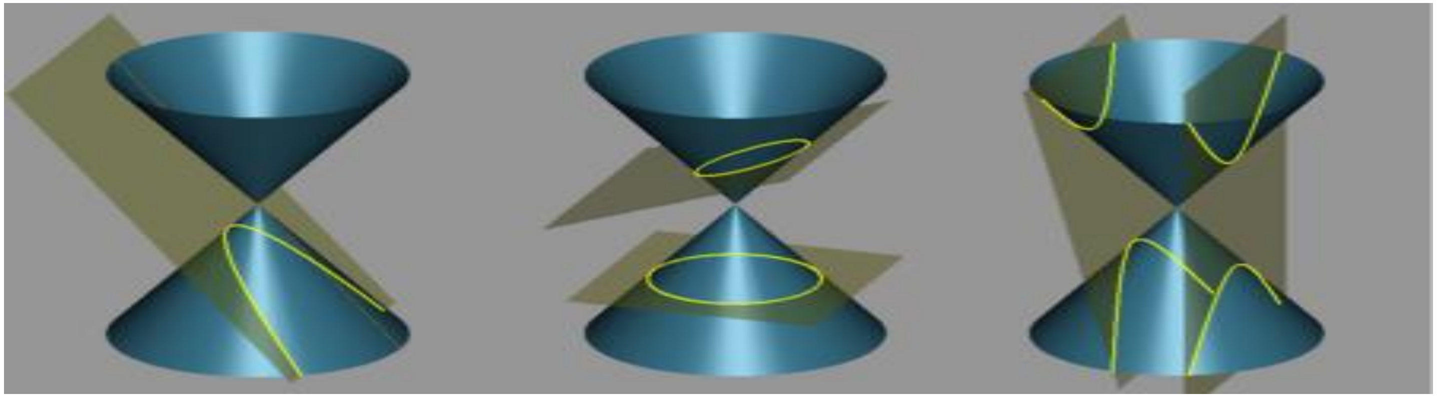
Piecewise B-Spline Curve of degree 3

Properties of B-Spline

1. Convex Hull Property
2. variation diminishing property.
3. Affine Invariance
4. local
5. piecewise polynomial

Why Introduces to NURBS?

- Disadvantage of B-Spline curve and Bezier Curve:
  - can’t accurately represent conic(圆锥曲线) curve except parabola(抛物线),
- NURBS (Non-Uniform Rational B-Spline) (非均匀有理B样条)
  - In order to find a mathematical method that could represent conic and conicoid(二次曲面) accurately.



抛物线　　椭圆(上)与圆(下)　　双曲线

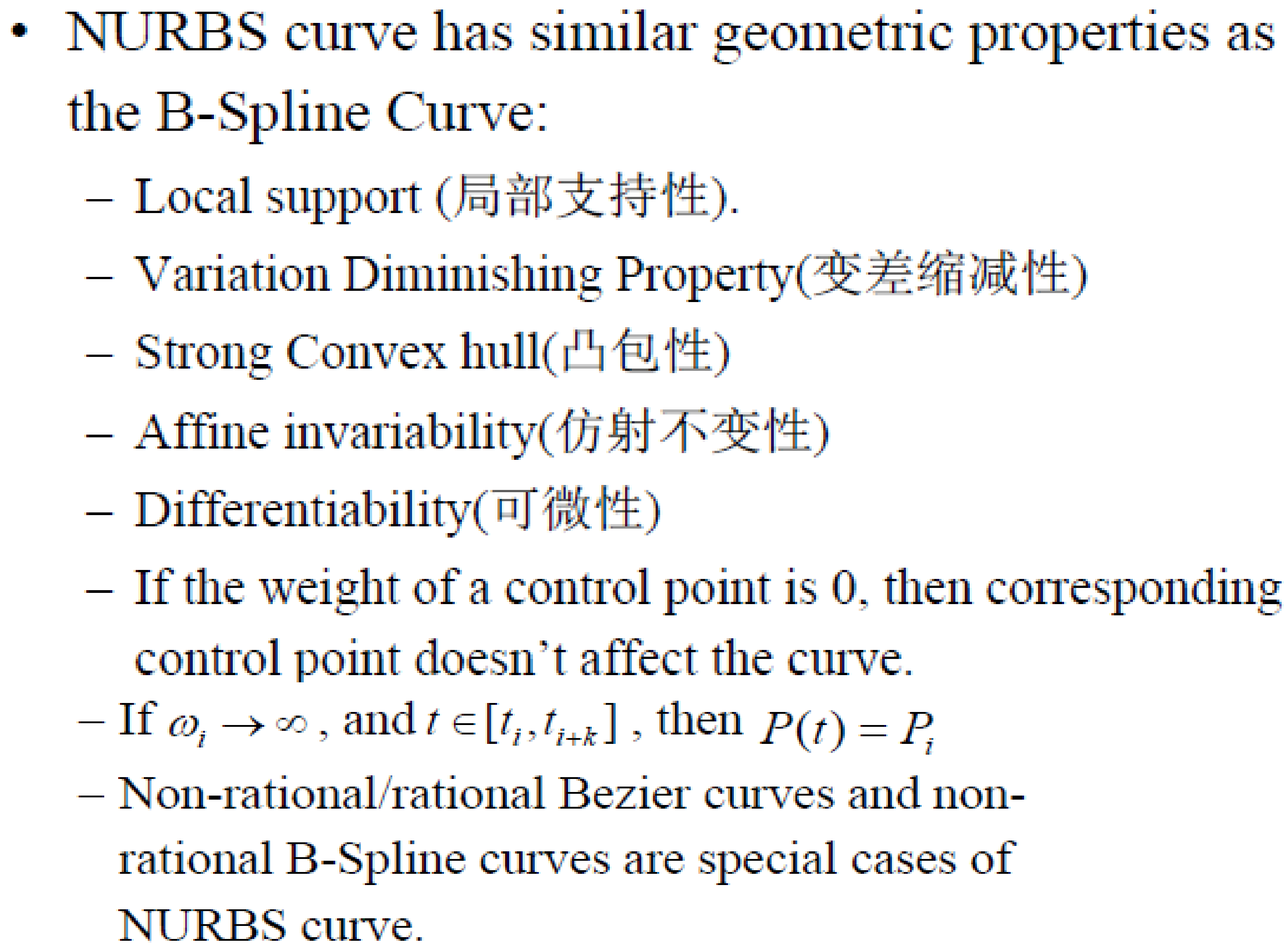
Definition of NURBS Curve

- Definition of NURBS curve.
  - NURBS Curves are defined by piecewise rational B-Spline polynomial basis function (分段有理B样条多项式基函数):

$$P(t)=\frac{\sum_{i=0}^{n}\omega_i P_i N_{i,k}(t)}{\sum_{i=0}^{n}\omega_i N_{i,k}(t)}=\sum_{i=0}^{n}P_i R_{i,k}(t)$$

$$R_{i,k}(t)=\frac{\omega_i N_{i,k}(t)}{\sum_{j=0}^{n}\omega_j N_{j,k}(t)}$$

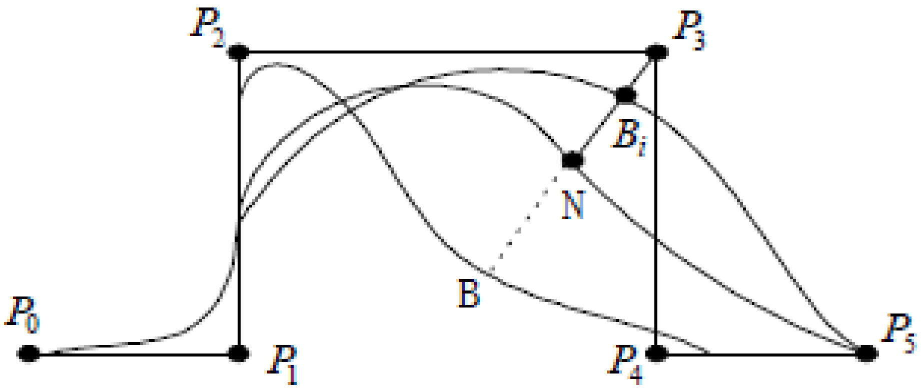
Propertise of NURBS Curve



- NURBS curve has similar geometric properties as the B-Spline Curve:
  - Local support (局部支持性).
  - Variation Diminishing Property(变差缩减性)
  - Strong Convex hull(凸包性)
  - Affine invariability(仿射不变性)
  - Differentiability(可微性)
  - If the weight of a control point is 0, then corresponding control point doesn't affect the curve.
  - If $\omega_i \to \infty$, and $t \in [t_i, t_{i+k}]$, then $P(t) = P_i$
  - Non-rational/rational Bezier curves and non-rational B-Spline curves are special cases of NURBS curve.

Geometric meaning of weights

- If ω_i increases/decreases, β also increases/decreases, and the curve is like pulling to/pushing away from point P_i.



NURBS describe a circle: example

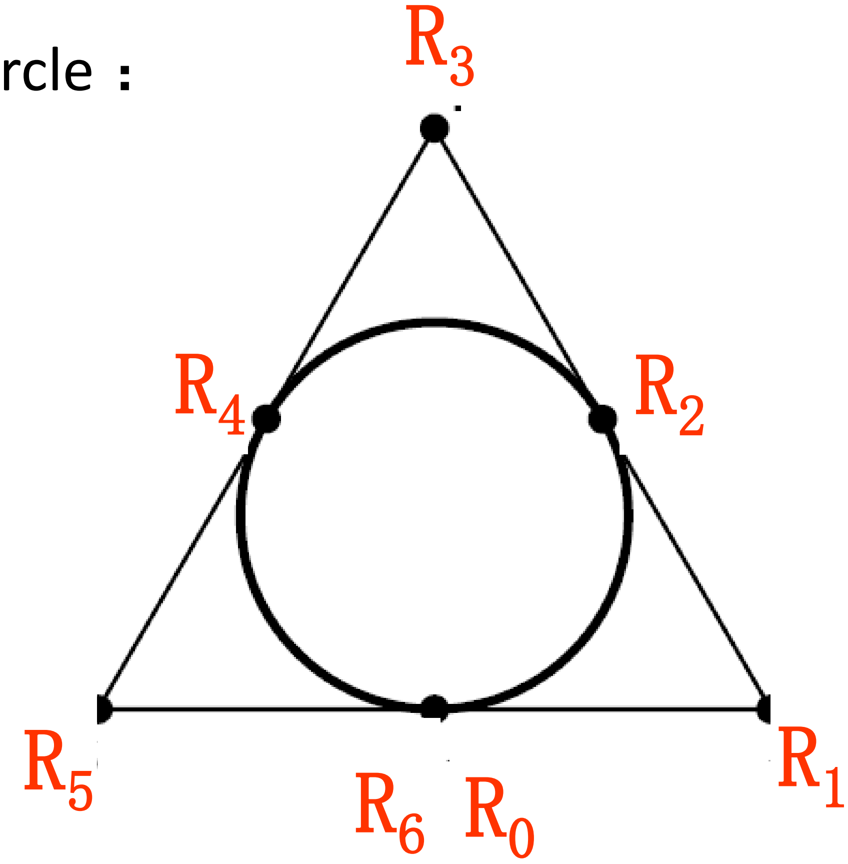
Three 120° circular arc describe circle：

u =[0 0 0 1 1 2 2 3 3 3]

$k = 3$

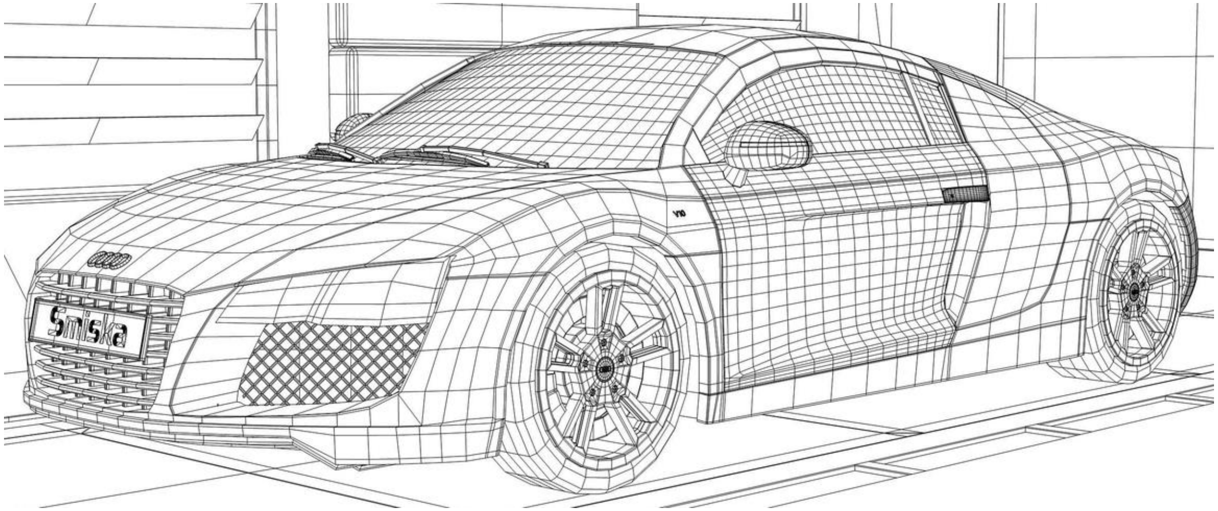
$[\omega_i] = [1, ½, 1 , ½, 1, ½, 1]$

控制顶点分布如右图所示



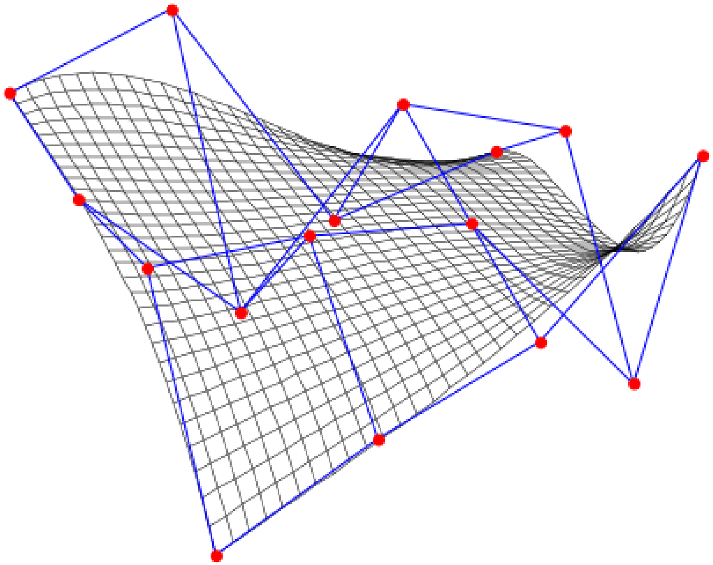
Outline

- Interpolation and Approximation
- Curve Modeling
    - Parametric curve
    - Cubic Hermite interpolation
    - Bézier curve
    - B-Spline
- Surface Modeling
    - Bézier surface



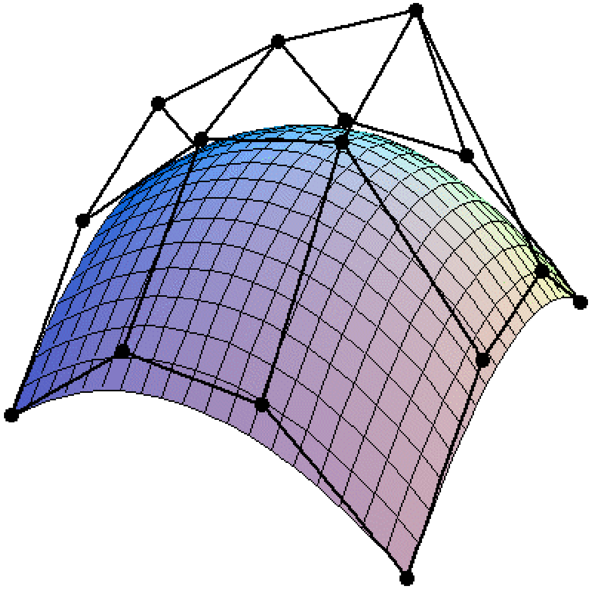
Bézier surface

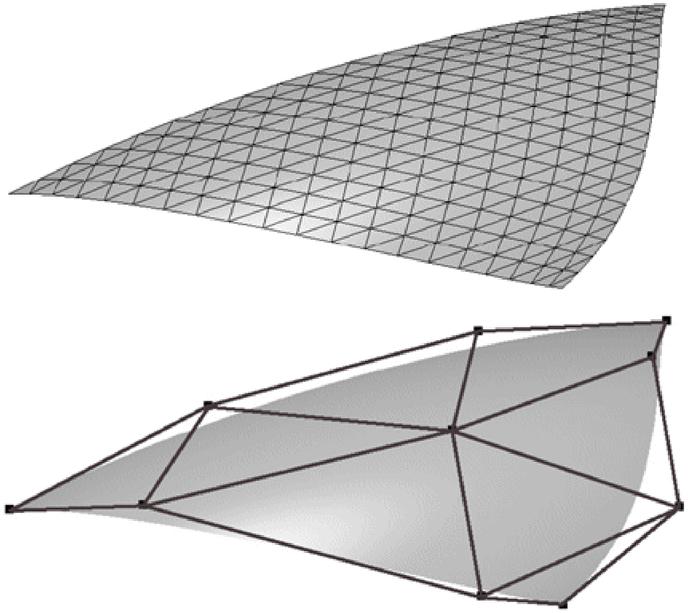
- Bézier surfaces are a species of mathematical spline used in computer graphics, computer-aided design, and finite element modeling.
- As with the Bézier curve, a Bézier surface is defined by a set of control points.



Bézier surface

- 曲面表示方法与参数区域的选择有着密切的关系，若选择矩形参数区域，一般采用张量积或布尔和形式来构造曲面；若选择三角形（即单纯形）参数区域，则要采用直接升阶构造方法来表示曲面。





The definition of Bézier surface in rectangular domain

- 当选定矩形参数区域$[0,1]\times[0,1]$后，则采用张量积（tensor product）方法把 Bézier curve 推广成 Bézier surface。
- 给定了$(n+1)(m+1)$个空间上顶点$P_{ij}(i=0,1,\dots,n;j=0,1,\dots,m)$，则称$n\times m$次参数曲面为$n\times m$次Bézier曲面，有：

$$P(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}B_{i,n}(u)B_{j,m}(v)P_{ij}\qquad (u,v\in[0,1])$$

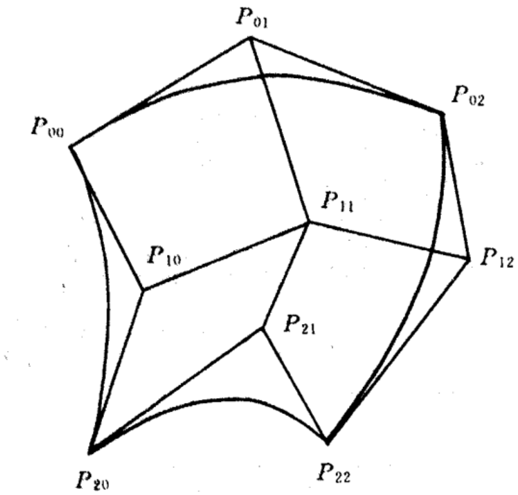
- 这里$B_{i,n}(u)$和$B_{j,m}(u)$为Bernstein基函数，依次用线段连接顶点$P_{ij}(i=0,1,\dots,n;j=0,1,\dots,m)$中相邻两顶点所形成的空间网格，称为Bézier曲面的特征网格。

Biquadratic Bézier surface

- 当n=m=2时，有biquadratic Bézier surface (双二次曲面)：

$$P(u,v)=\sum_{i=0}^{2}\sum_{j=0}^{2}B_{i,2}(u)B_{j,2}(v)P_{ij}\qquad (u,v\in[0,1])$$

- 该曲面的4条边界曲线都是抛物线，实际上其特征网格上的9个顶点，其中有8个边界顶点来确定4条边界曲线。
- 只有一个顶点P_{11}可以用来控制曲面的形状，因此曲面的凹凸可以通过控制P_{11}来直观控制。



Bicubic Bézier surface

- 当n=m=3时，有bicubic Bézier surface (双三次曲面)：

$$P(u,v)=\sum_{i=0}^{3}\sum_{j=0}^{3}B_{i,3}(u)\,B_{j,3}(v)\,P_{ij} \qquad (u,v\in[0,1])$$

- 有矩阵表示：

$$P(u,v)=[B_{0,3}(u),B_{1,3}(u),B_{2,3}(u),B_{3,3}(u)]$$

$$\times\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix}\begin{bmatrix} B_{0,3}(v) \\ B_{1,3}(v) \\ B_{2,3}(v) \\ B_{3,3}(v) \end{bmatrix} \qquad (u,v\in[0,1])$$

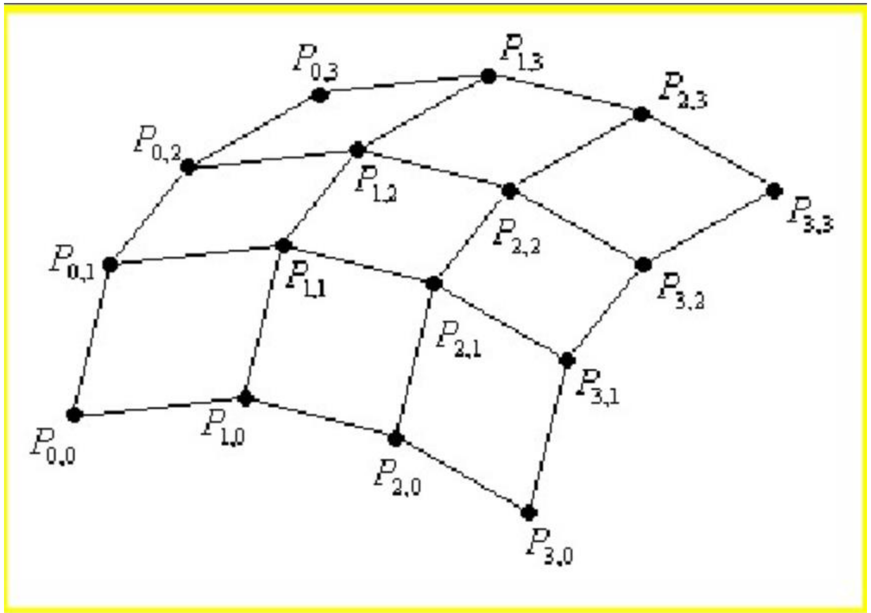
- 进一步简化： $P(u,v)=UBPB^TV^T$

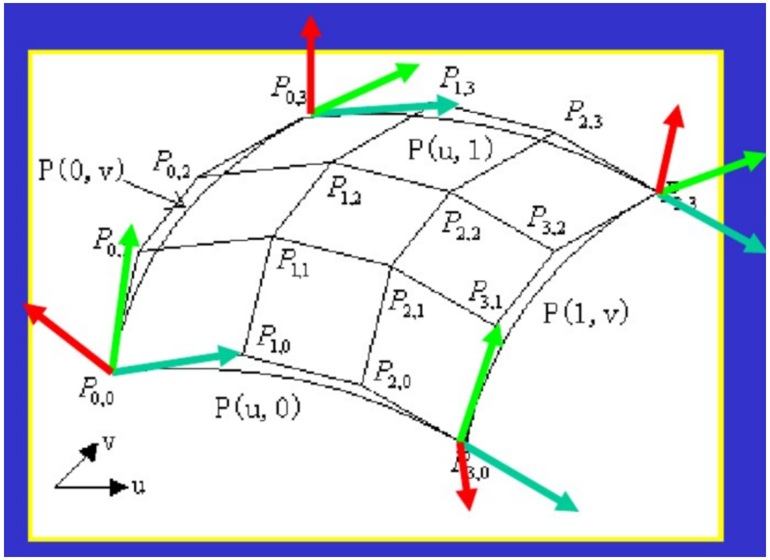
- 其中：

$$U^T=\begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix},\quad B=\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix},\quad P=\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix}$$

Bicubic Bézier surface

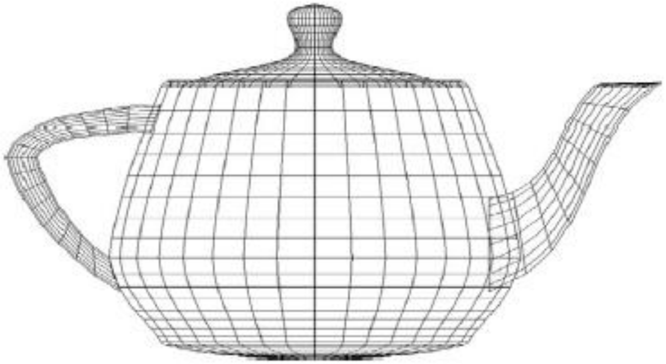
- 曲面的4条边界都是三次Bézier曲线，由周边12个特征网格的顶点来确定。可通过调整内部4个顶点P_{11}，P_{12}，P_{21}，P_{22}的位置来控制曲面内部的形状。





Utah Teapot

- 计算机图形学中最著名的数据集之一
- 通常用到的是由363个三维顶点定义的32张Bézier曲面片



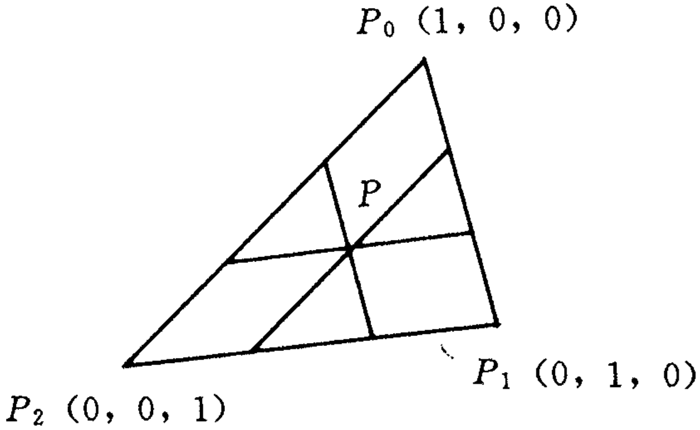
Bézier surface in triangular domain

- 当采用三角形参数区域$\{(u, v, w)|u, v, w \geq 0,\ u+v+w=1\}$后，需要选择B-网的方法来构造Bézier 曲面。
- 对于不共线的三个顶点P_0，P_1，P_2就可以形成一个三角形，因此三角形组成的平面上任意一点P可以表示成：

$$P(u, v, w) = uP_0 + vP_1 + wP_2 \qquad (u+v+w=1)$$

- 其中：(u, v, w) 称为$P(u, v, w)$关于P_0，P_1，P_2的重心坐标。
- 当$0 \leq u+v+w \leq 1$时，$P(u, v, w)$位于

  P_0，P_1，P_2组成的三角形之内。



The definition of Bézier surface in triangular domain

- 定义：参数曲面为三角域上的n次Bézier曲面：

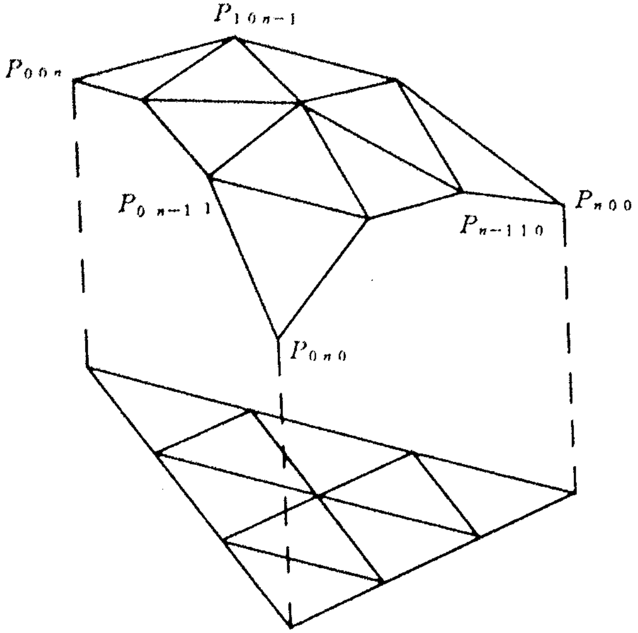
$$P(u, v, w) = \sum_{i+j+j=n} P_{ijk} B^n_{ijk}(u, v, w)$$

$$(u, v, w \geqslant 0;\ u+v+w=1)$$

- 其中：

$$B^n_{ijk}(u, v, w) = \frac{n!}{i!\ j!\ k!} u^i v^j w^k$$

$$(i+j+k=n;\ u+v+w=1)$$

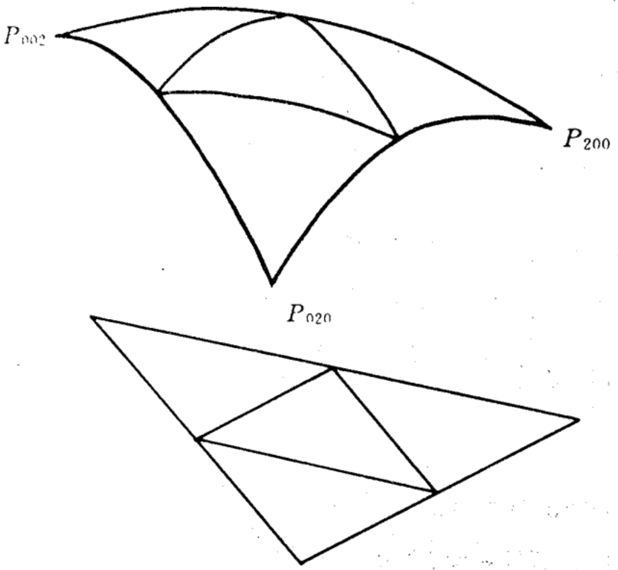


Quadratic Bézier surface

- 当n=2时，可以直接写出二次Bézier曲面

$$
\begin{aligned}
P(u, v, w) &= \sum_{i+j+k=n} P_{ijk} B_{ijk}^2 (u, v, w) \\
&= u^2 P_{200} + v^2 P_{020} + w^2 P_{002} + 2uvP_{110} + 2uwP_{101} + 2vwP_{011}
\end{aligned}
$$

- 从上式中可以看出，二次Bézier曲面完全由6个边界顶点确定，因此，其形状完全由边界曲线所控制。

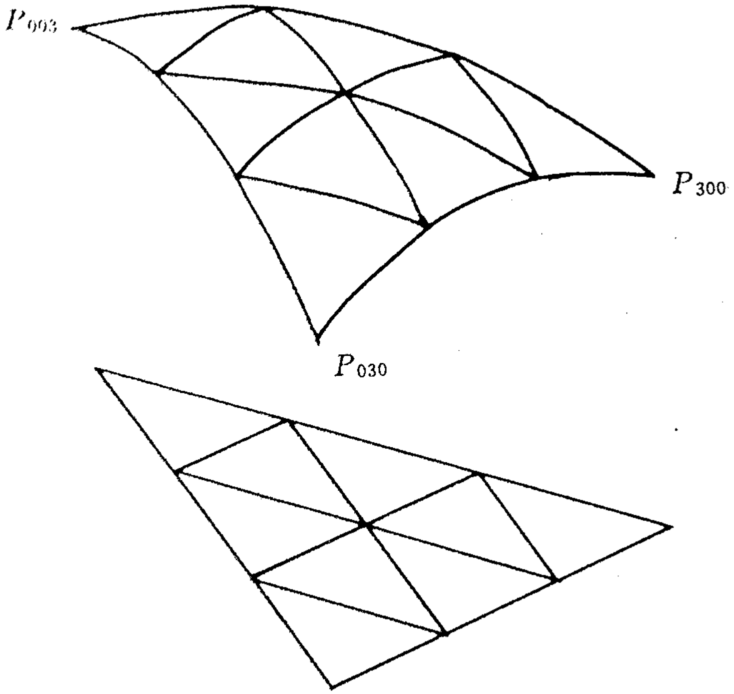


Cubic Bézier surface

- 三次Bézier曲面为：

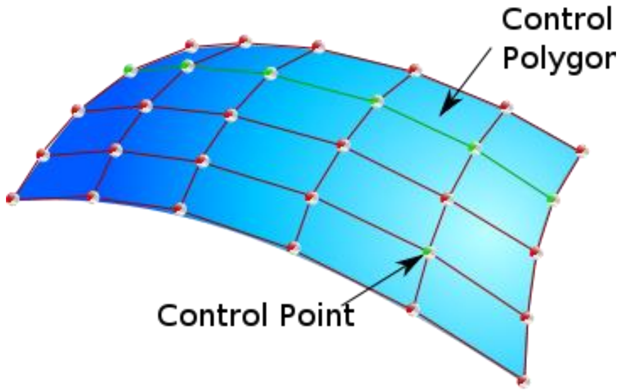
$$
\begin{aligned}
P(u, v, w) &= \sum_{i+j+k=3} P_{ijk}B^3_{ijk}(u, v, w) \\
&= u^3P_{300} + v^3P_{030} + w^3P_{003} + 3u^2vP_{210} + 3uv^2P_{120} \\
&\quad + 3u^2wP_{201} + 3uw^2P_{102} + 3v^2wP_{021} + 3vw^2P_{012} + 6uvwP_{111}
\end{aligned}
$$

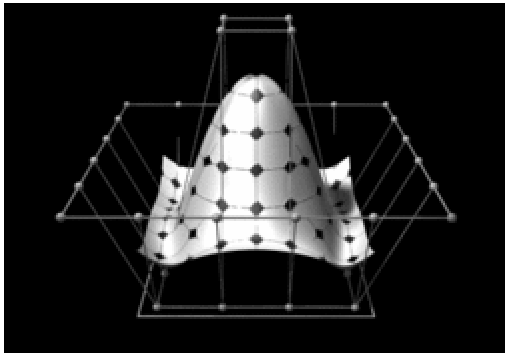
- 若选定了三次Bézier曲面的9个边界顶点，其内部形状就可由一个控制顶点P_{111}来交互改变。

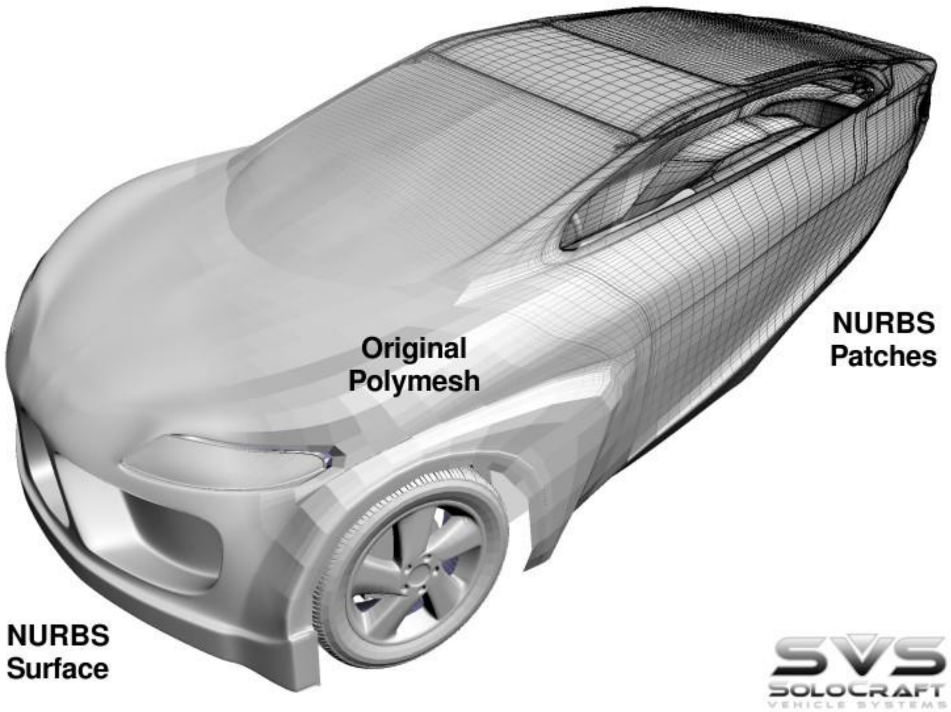


Non-uniform rational B-spline (NURBS) surface

- Non-uniform rational basis spline (NURBS) is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces.





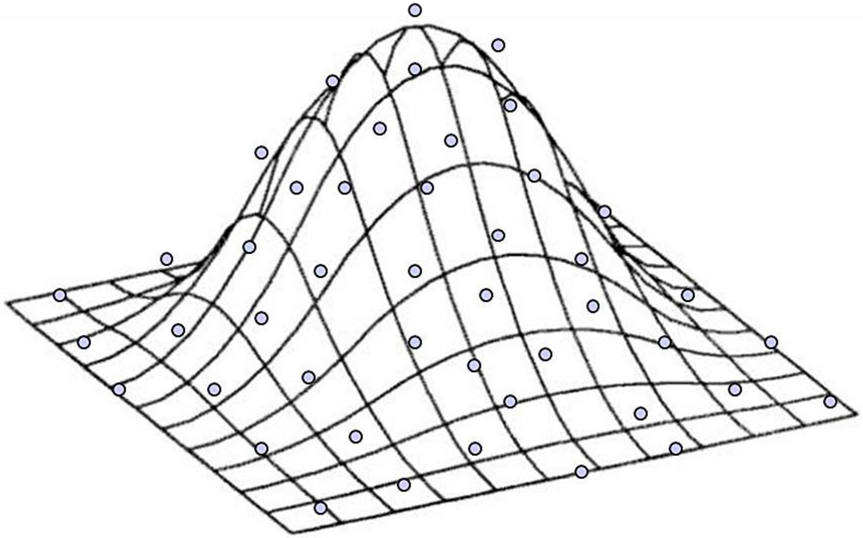


Least square surface

- If you have a data set (x_1, y_1, z_1), (x_2, y_2, z_2) , ... , (x_n, y_n, zn) and the best surface $z = f(x, y)$ should be with the property as follows

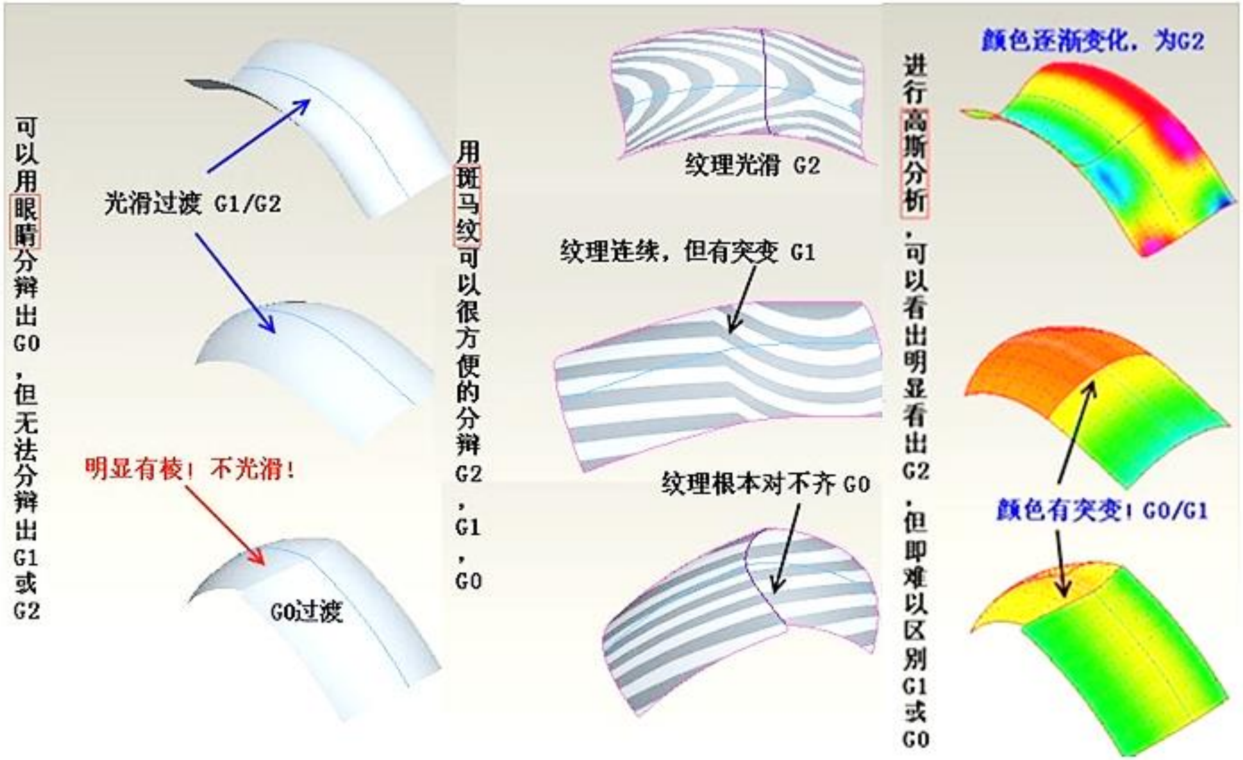
Minimum Least Square error

$$E = \sum_{i=1}^{n} (f(x_i, y_i) - z_i)^2$$



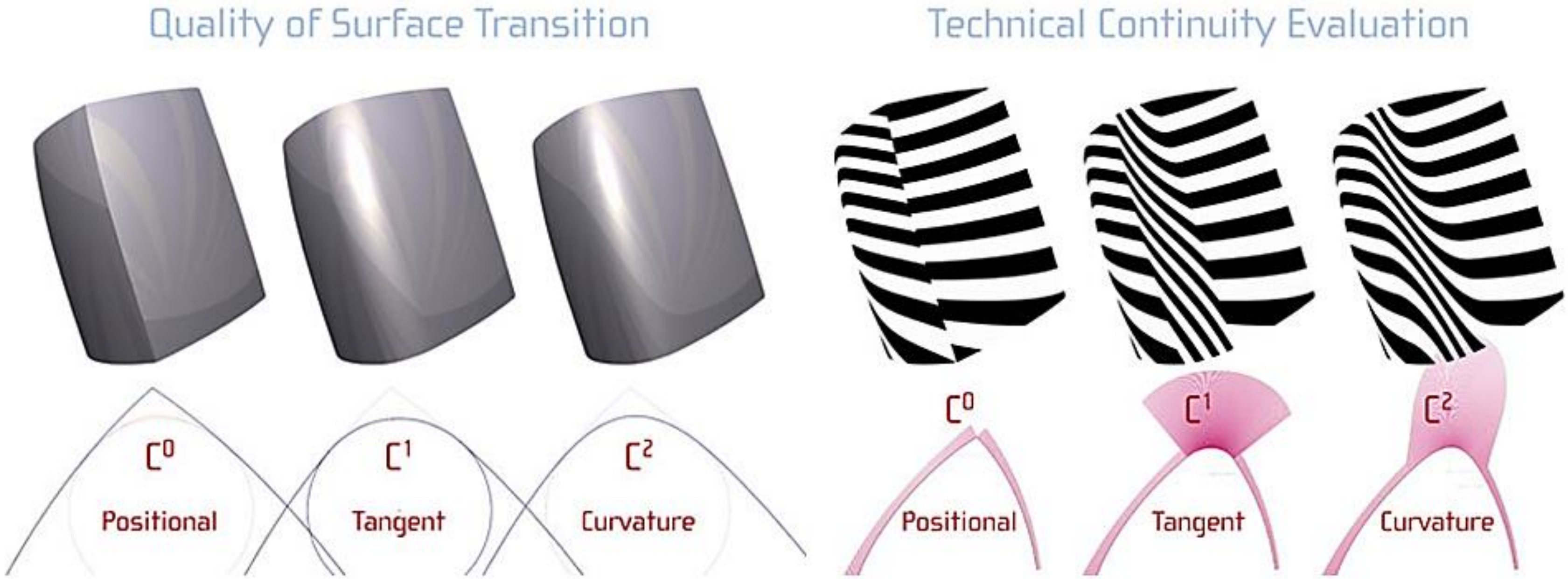
The continuity of the curve and surface

- G^n是两个几何对象间的实际连续程度，用于表示实际物理连续性；而 C^n 是实际物理连续性的数学表达，几何连续性G^n没有数学（参数）连续性C^n严格。 G^n的连续性是独立于表示（参数化）的。


可以用眼睛分辩出G0，但无法分辩出G1或G2
光滑过渡 G1/G2
明显有棱！不光滑！
G0过渡
用斑马纹可以很方便的分辩G2，G1，G0
纹理光滑 G2
纹理连续，但有突变 G1
纹理根本对不齐 G0
进行高斯分析，可以看出明显看出G2，但即难以区别G1或G0
颜色逐渐变化，为G2
颜色有突变！G0/G1

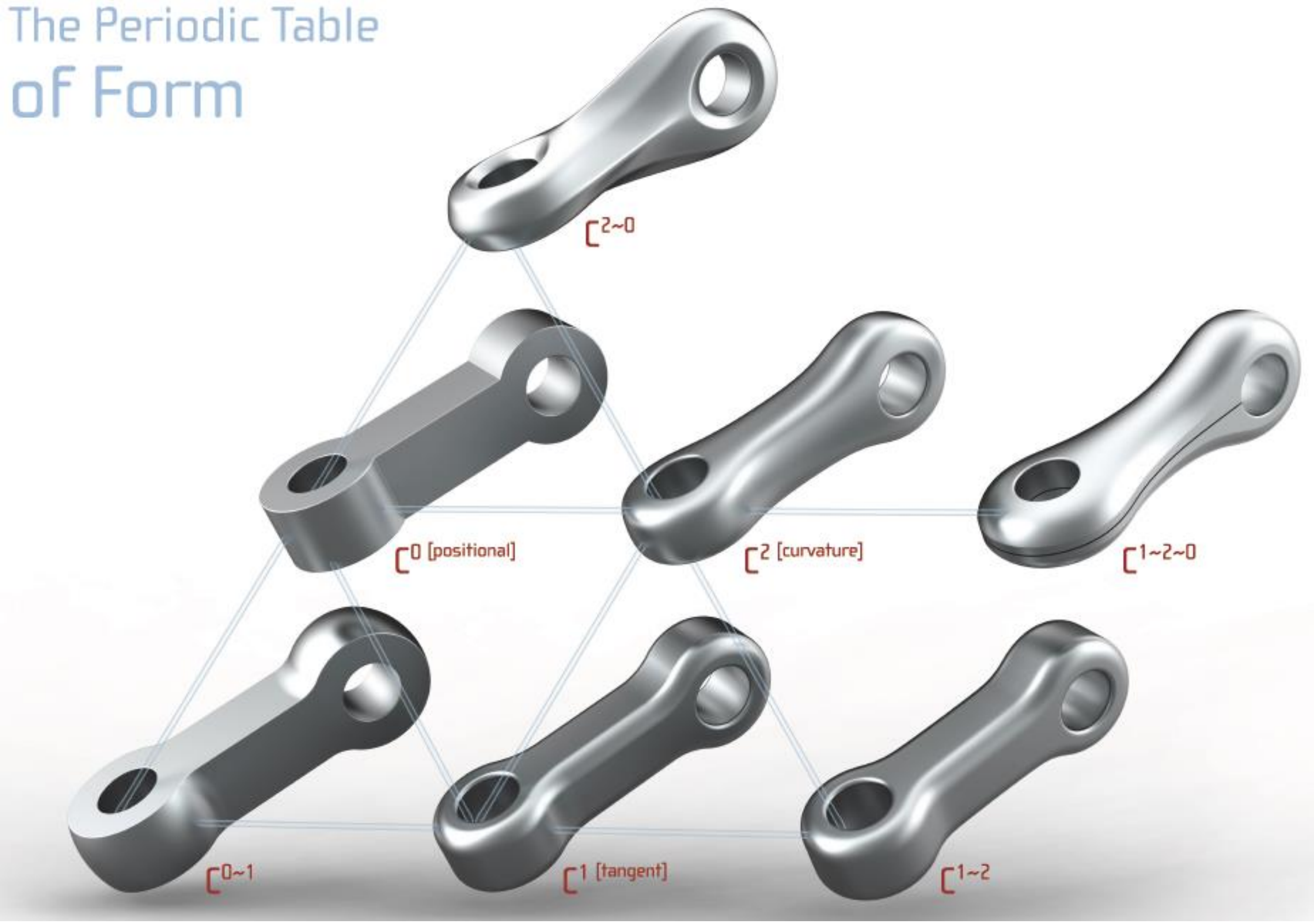
The continuity of the curve and surface

- C^0 –意味着两个相邻段间存在一个公共点（即两个段相连）。
- C^1 –对应于曲线方程的1阶导数，即意味着有一个公共点，并且多项式的一阶导数（即切向矢量）是相同的。
- C^2 –对应于曲线方程的2阶导数，意味着一阶导数和二阶导数都相同。



The continuity in real world

Simplified cases


The Periodic Table
of Form
$C^{2\sim0}$
C^{0} [positional]
C^{2} [curvature]
$C^{1\sim2\sim0}$
$C^{0\sim1}$
C^{1} [tangent]
$C^{1\sim2}$