Malaysian Society for Numerical Methods

PPM-015-08-16112017

MSNM live session series on Computational Mechanics and Numerical Methods

Curves and Surfaces for CAD Systems

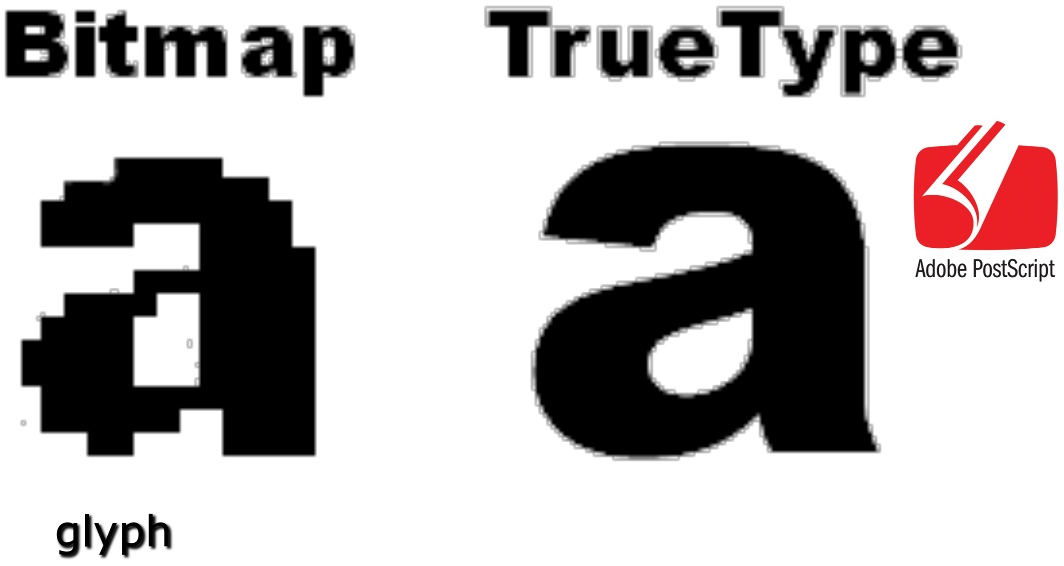
R.U. Gobithaasan

Faculty of Ocean Engineering Technology & Informatics,
University Malaysia Terengganu

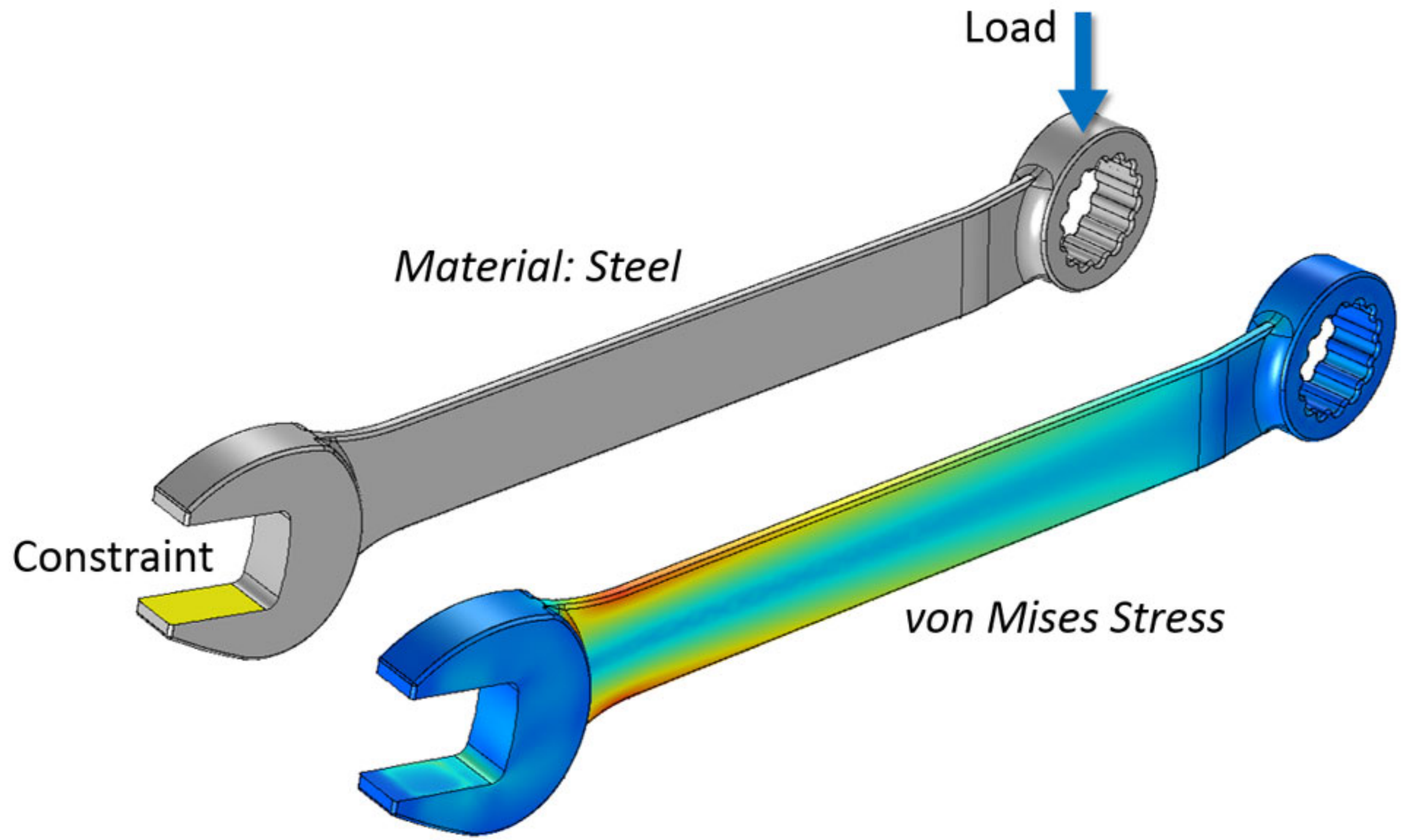
8th Session | 14th August 2020 (Fri) at 10am

...via FB live / webex platform

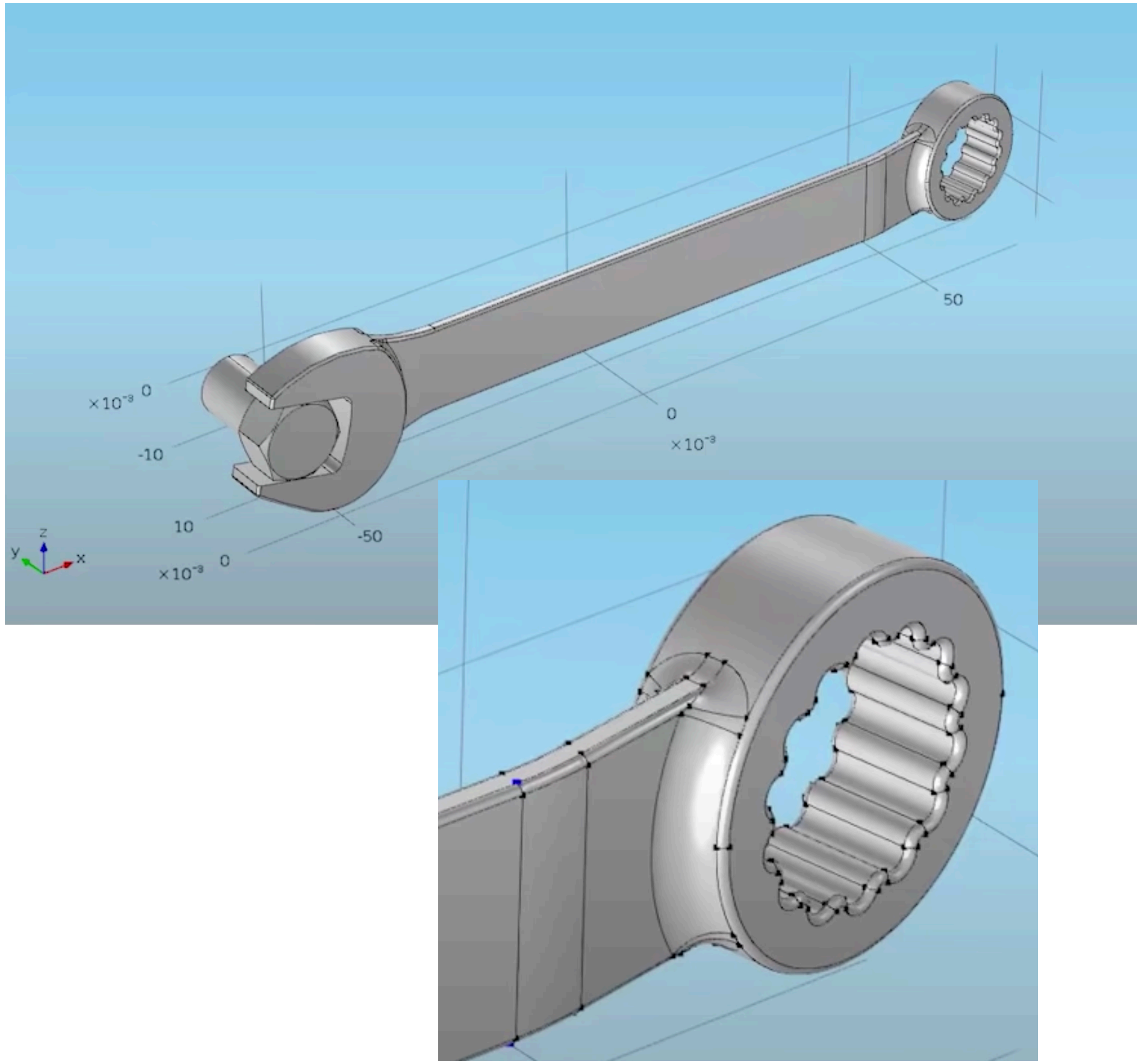
Table of Content

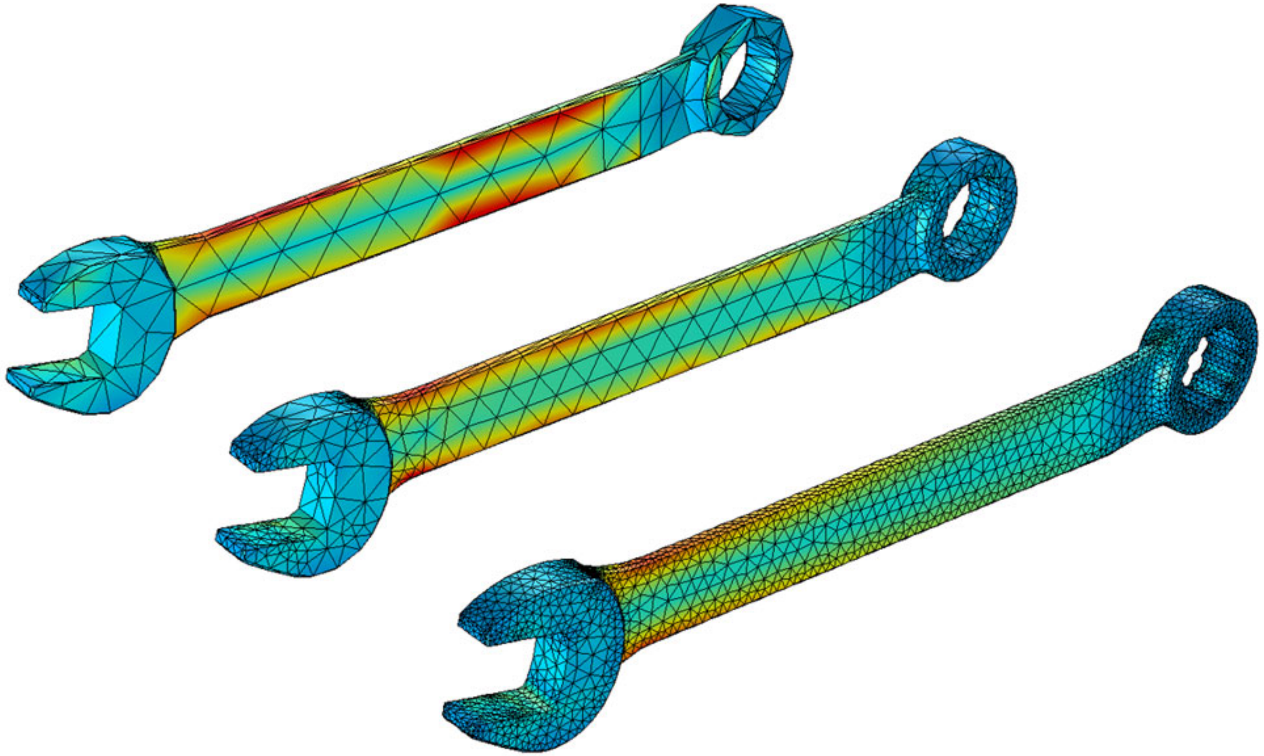

Bitmap
TrueType
Adobe PostScript
glyph

MOTIVATION


Load
Material: Steel
Constraint
von Mises Stress

CAD: Exact representation of shape & structure
CAE: Discretization of the shape (approximation)



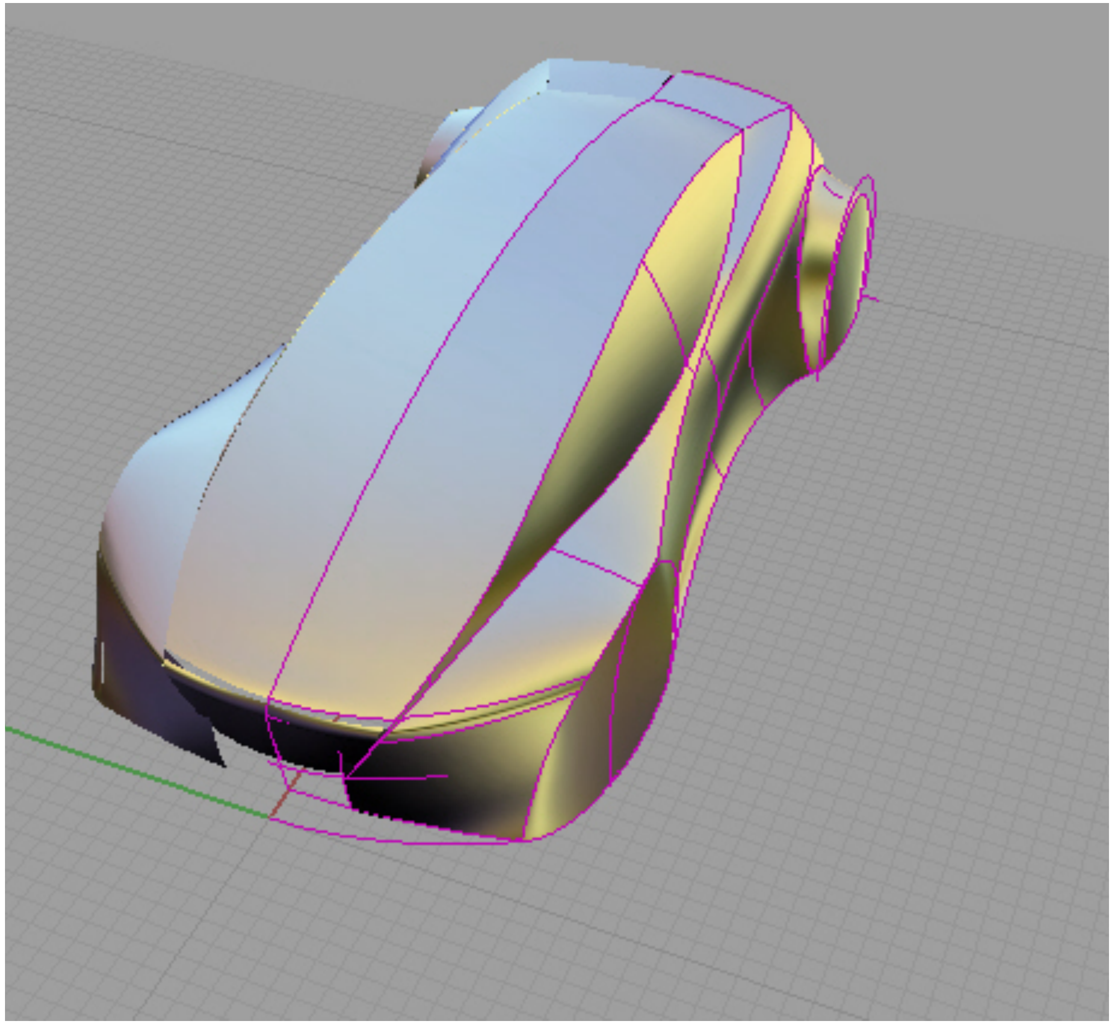


https://www.youtube.com/watch?v=GjSdI9lxpy8

https://www.comsol.com/multiphysics/mesh-refinement

What's in a surface?

Interior keylines

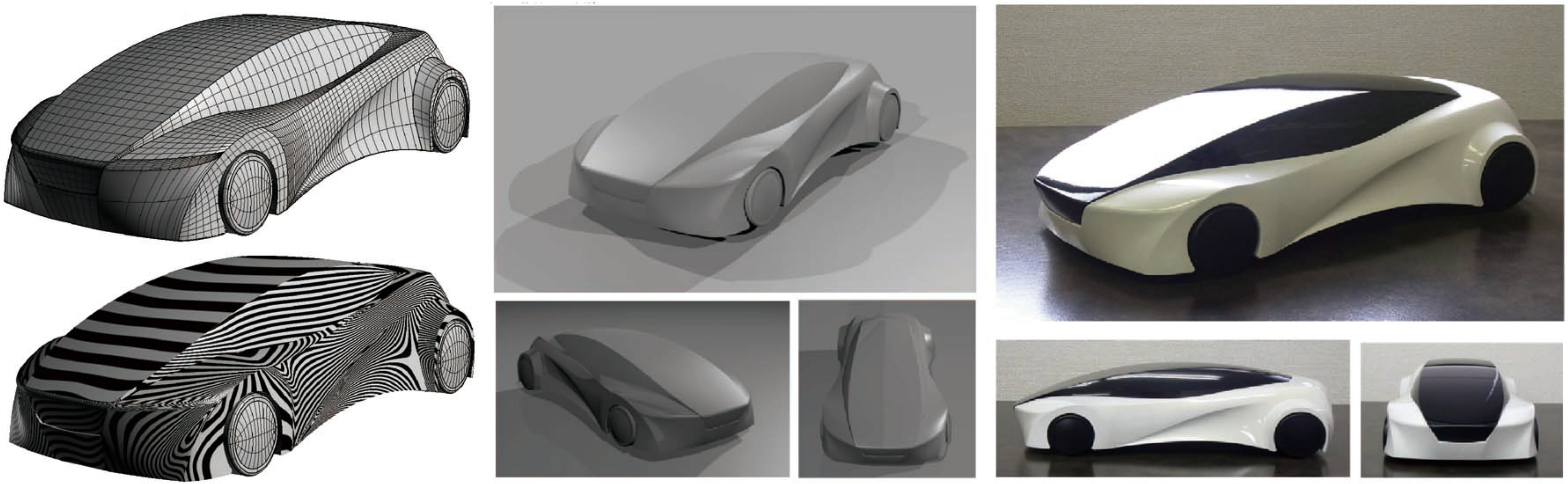


https://www.carbodydesign.com/gallery/2013/04/exclusive-daf-xf-design-story/13/

KT Miura, D Shibuya, RU Gobithaasan, S Usuki, Designing Log-aesthetic Splines with G2 continuity, Computer-Aided Design and Applications 10 (6), 1021-1032

Whats in a surface?

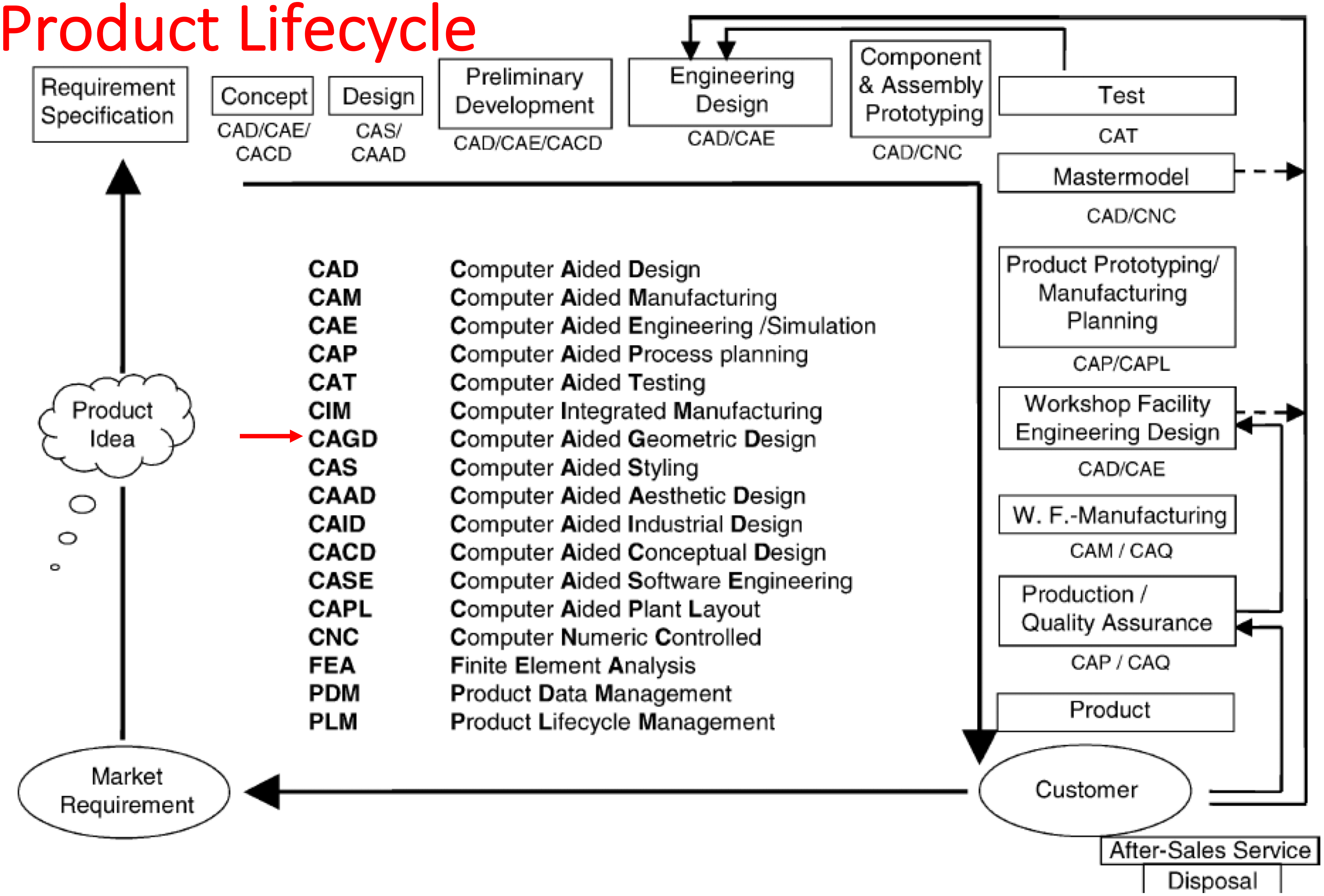
Rhinoceros®
NURBS Modeling for Windows



(a) Isoparametric lines and zebra mapping (b) Rendering (c) Mock-up

KT Miura, D Shibuya, RU Gobithaasan, S Usuki, Designing Log-aesthetic Splines with G2 continuity, Computer-Aided Design and Applications 10 (6), 1021-1032

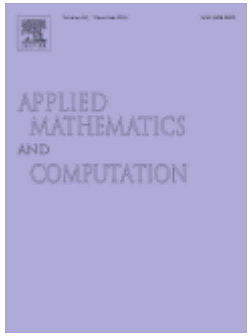
CA-x: Product Lifecycle

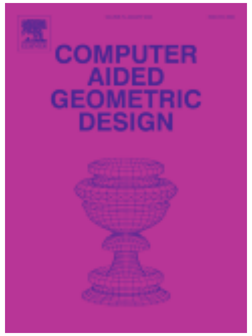


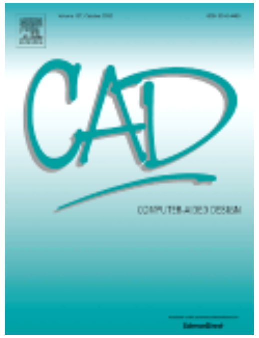
C. W. Dankwort, R. Weidlich, B. Guenther, and J. E. Blaurock. Engineers' Cax education{it's not only CAD. Computer-Aided Design, 36(14):1439-1450, 2004.

Computer Aided Geometric Design (CAGD)

Rekabentuk Geometri Berbantu Komputer (RGBK)

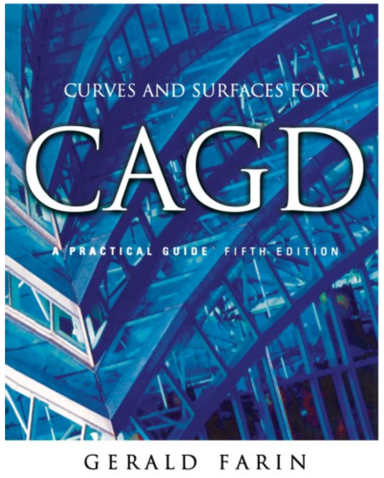


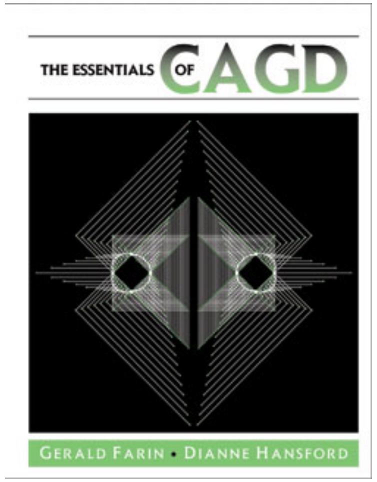


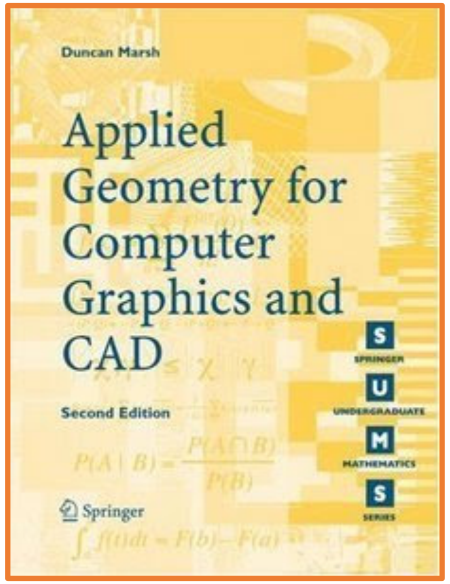


Keyword:

- Approximation Theory;
- Computational Geometry;
- Differential Geometry;
- Curve, Surface, and Solid;
- Computer-Aided-x (CAx);







CAGD: Malaysia

- USM (School of Math. Sc.); spearheaded by Prof. Dato Hasan Said / Prof. S. L. Lee
- CAGD Experts served at USM:
  - T.N.T Goodman
  - Keith Unsworth
- USM Supervisor:
  - Prof J. M. Ali
    - Supervised by Prof Alan Ball (CAD of British Airways)

A Generalized Ball Curve and Its Recursive Algorithm

H. B. SAID
Universiti Sains Malaysia

The use of Bernstein polynomials as the basis functions in Bézier's UNISURF is well known. These basis functions possess the shape-preserving properties that are required in designing free form curves and surfaces. These curves and surfaces are computed efficiently using the de Casteljau Algorithm. Ball uses a similar approach in defining cubic curves and bicubic surfaces in his CONSURF program. The basis functions employed are slightly different from the Bernstein polynomials. However, they also possess the same shape-preserving properties. A generalization of these cubic basis functions of Ball, such that higher order curves and surfaces can be defined and a recursive algorithm for generating the generalized curve are presented. The algorithm could be extended to generate a generalized surface in much the same way that the de Casteljau Algorithm could be used to generate a Bézier surface.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*curve, surface, solid and object representations*

General Terms: Algorithms

Additional Key Words and Phrases: Ball curves, Bernstein polynomials, Bézier curves, computer-aided geometric design, curves and surfaces

1. INTRODUCTION

The work of Bézier [4] on computer-aided design of free-form curves and surfaces has been widely discussed (see [6]). Bézier uses Bernstein polynomials as the basis functions of his scheme, and the scheme approximates a set of control points. These control points form a control polygon (polyhedron) for its curve (surface). The shape of the curve (surface) mimics the shape of the control polygon (polyhedron).

In computer-aided geometric design and computer graphics, efficient algorithms for generating curves or surfaces are important. One of the methods for generating a Bézier curve or surface is by using a recursive algorithm known as de Casteljau Algorithm (see [5] and [8]).

Ball [1–3] uses different basis functions to define his lofting surface program CONSURF at the British Aircraft Corporation. The method is analogous to the

Author's address: School of Mathematical and Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia.

ACM Transactions on Graphics, Vol. 8, No. 4, October 1989, Pages 360–371.

Profesor Emeritus Datuk Dr Hasan Said Naib

Computer-Aided Design
Volume 23, Issue 8, October 1991, Pages 554-560

Properties of generalized Ball curves and surfaces

T.N.T. Goodman **, H.B. Said *

Show more

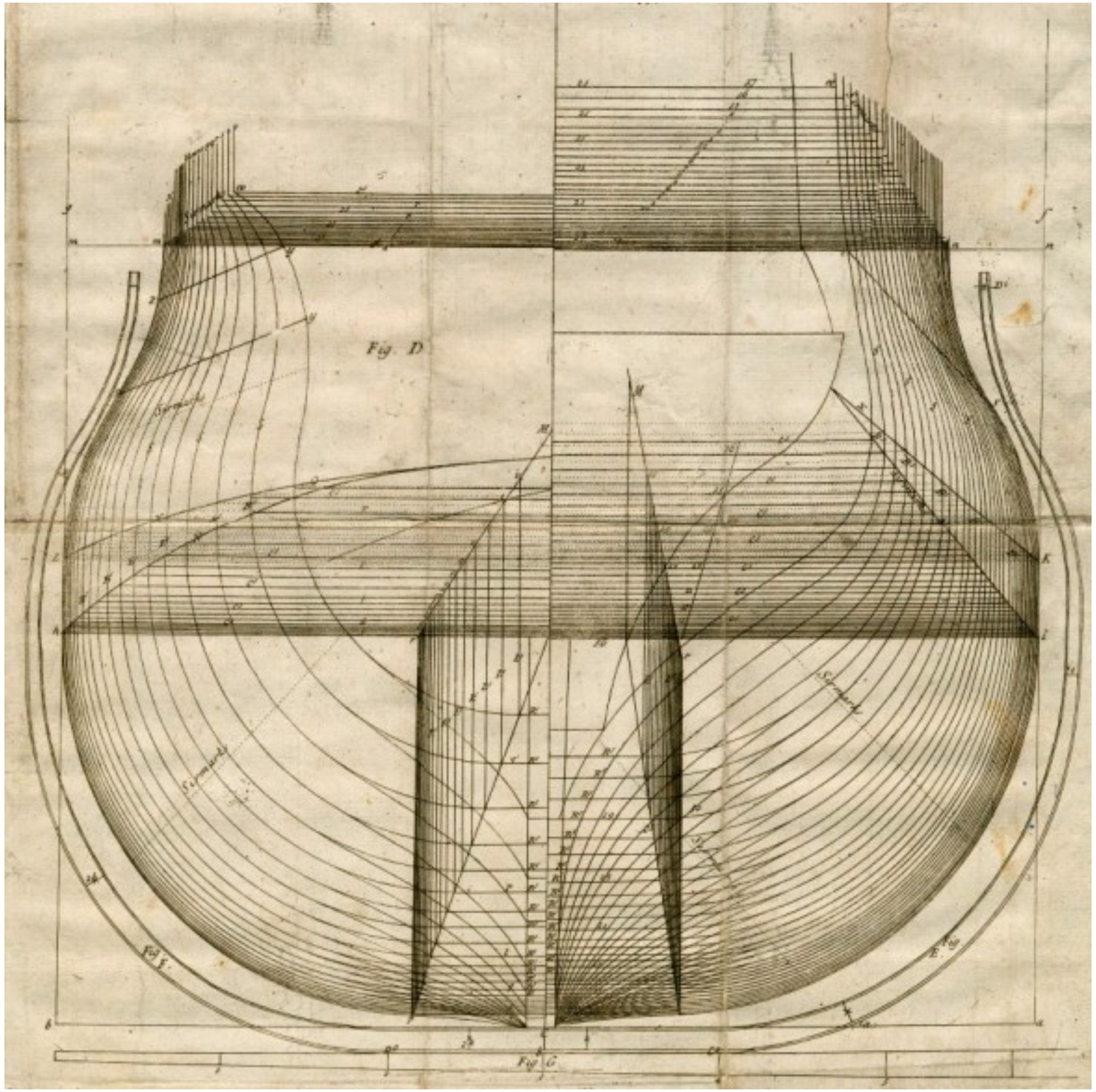
https://doi.org/10.1016/0010-4485(91)90056-3

Get rights and content

Abstract

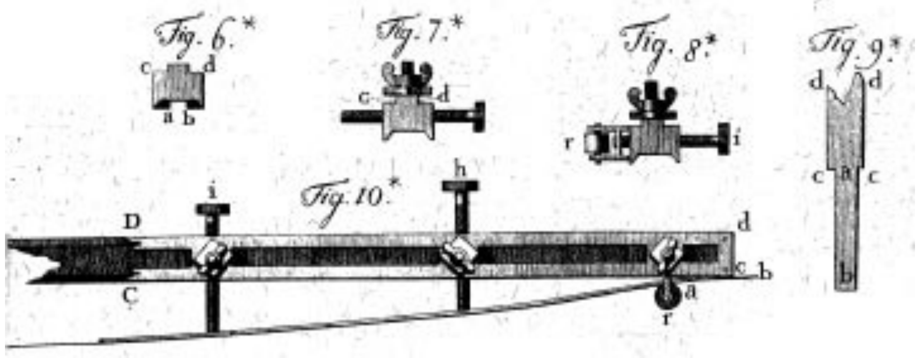
It is shown that the generalized Ball representation for a polynomial curve is much better suited to degree raising and lowering than the Bézier representation. The generalized Ball basis is then extended to polynomial surfaces over a triangle, and recursive algorithms for evaluation and degree raising are given.

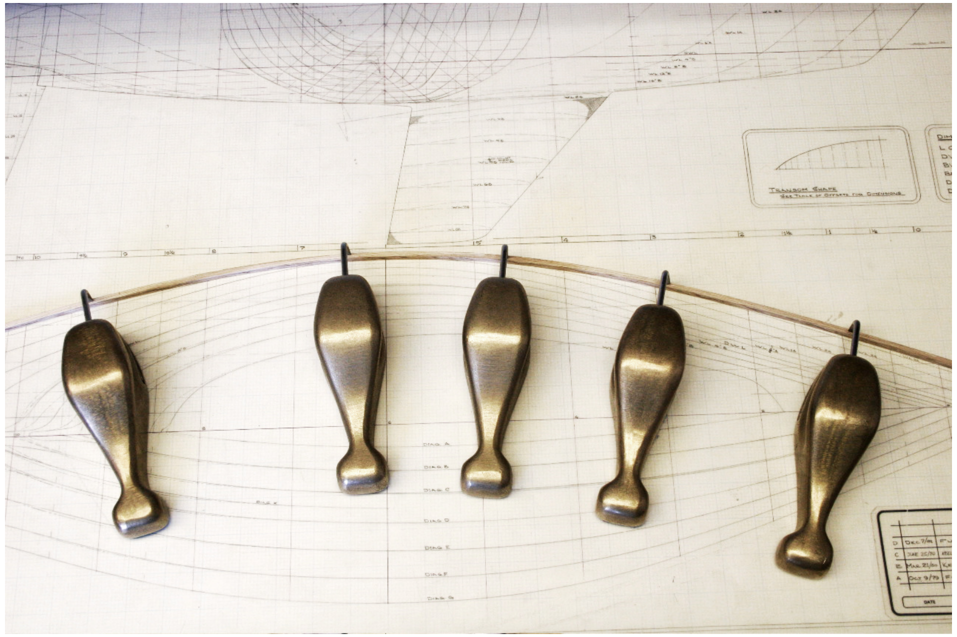
CAGD: HISTORY



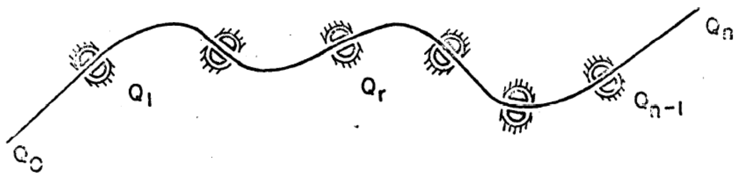
1700s; A ship's body plan comprised of superimposed construction geometry.

Ducks & Mechanical splines



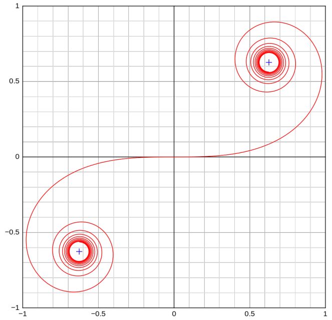


- *Hooked weights, called "ducks," accurately traces a spline.*
- Schoenberg *1946: polynomial spline of order k ≈ **Minimum Energy Curve (MEC)**

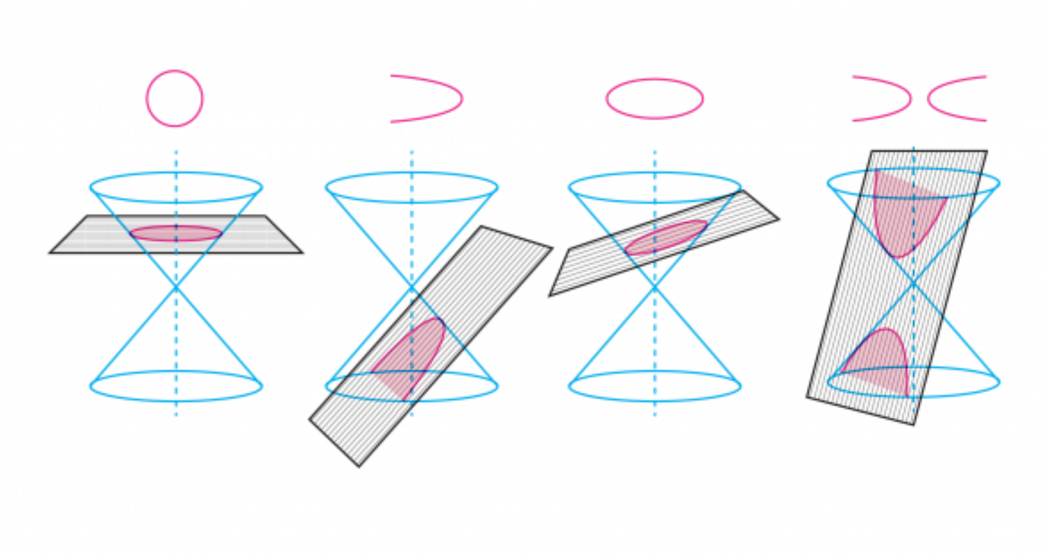


$$E[\kappa(s)] = \int_0^l \kappa(s)^2 \, ds$$

French curves

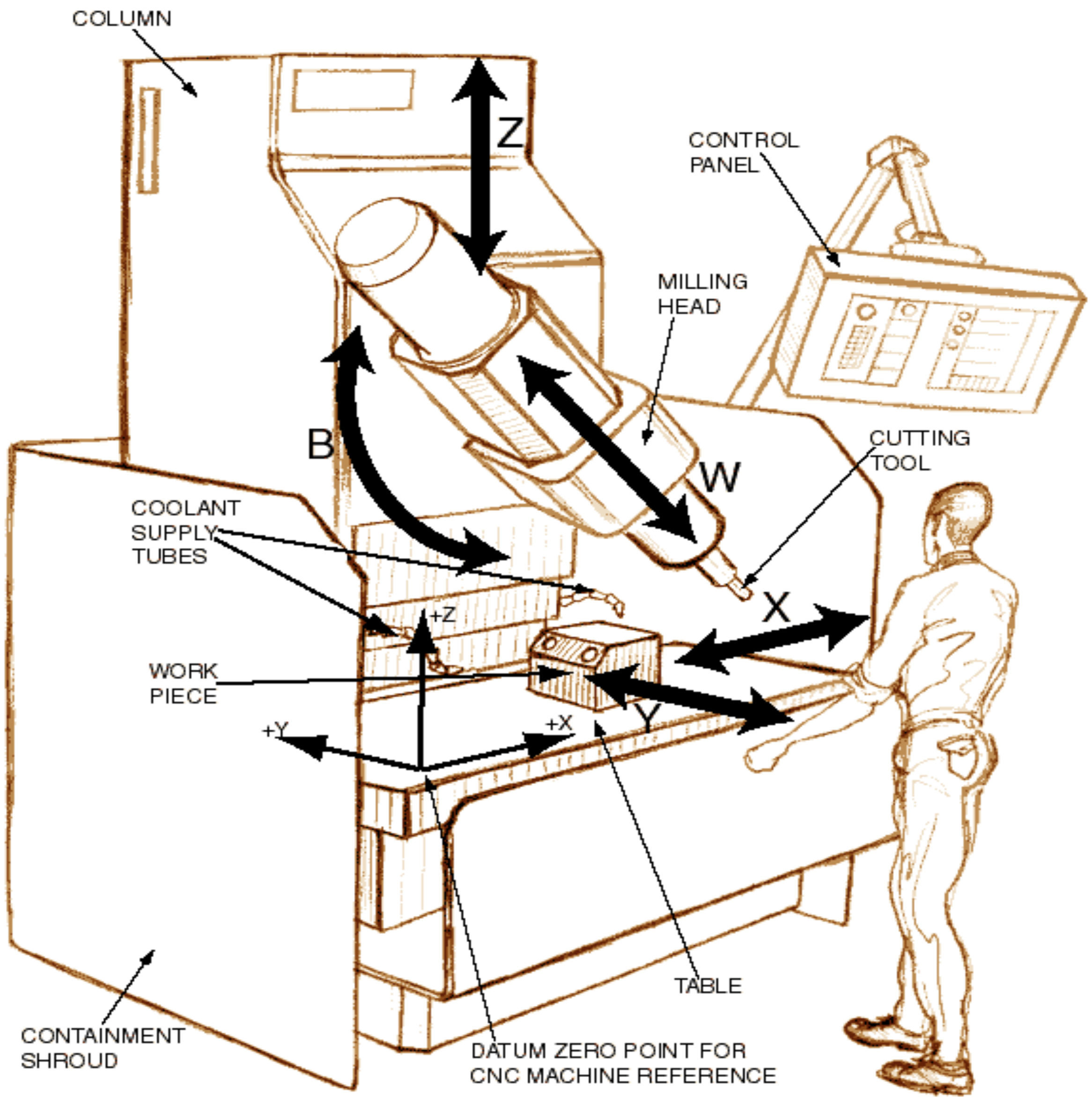


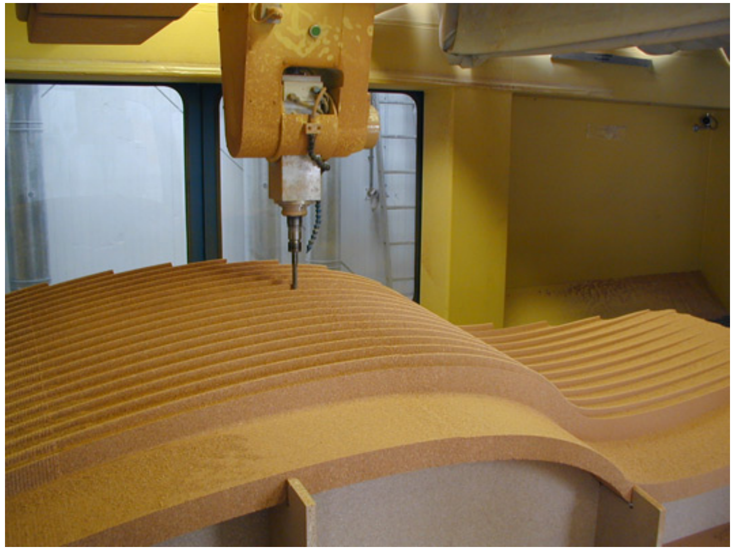
- Set of French Curves devised by the German kinematician and geometer Ludwig Burmester.
- The shapes are segments of clothoid curves & conic section



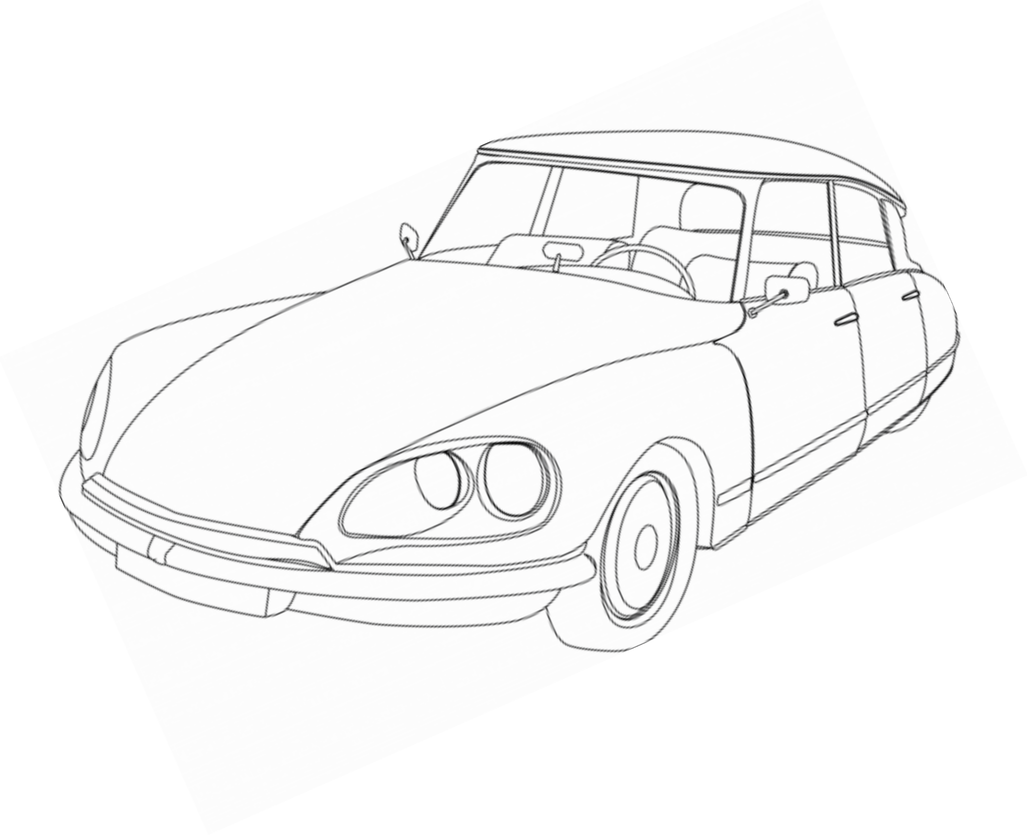
- Conic sections: e.g. aircraft design (Consurf for British Aircraft Corporation developed by A.A. Ball used birational cubic)

CAGD Precursor: Numerical Controls (1950s)



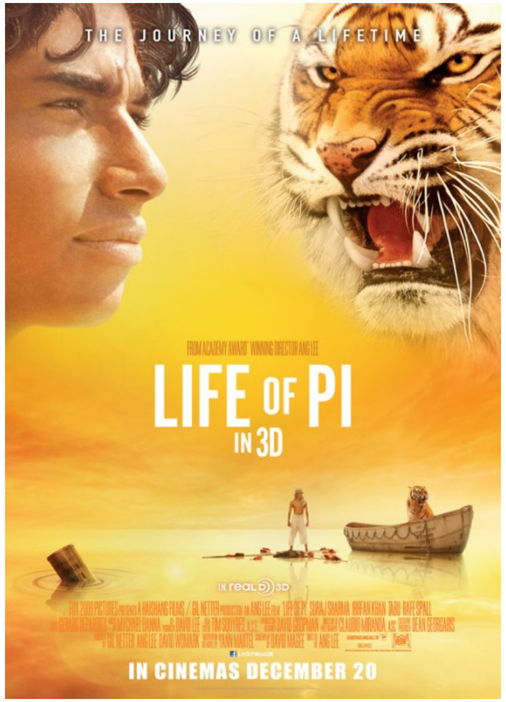


CAGD Precursor (past and present): Automobile industry



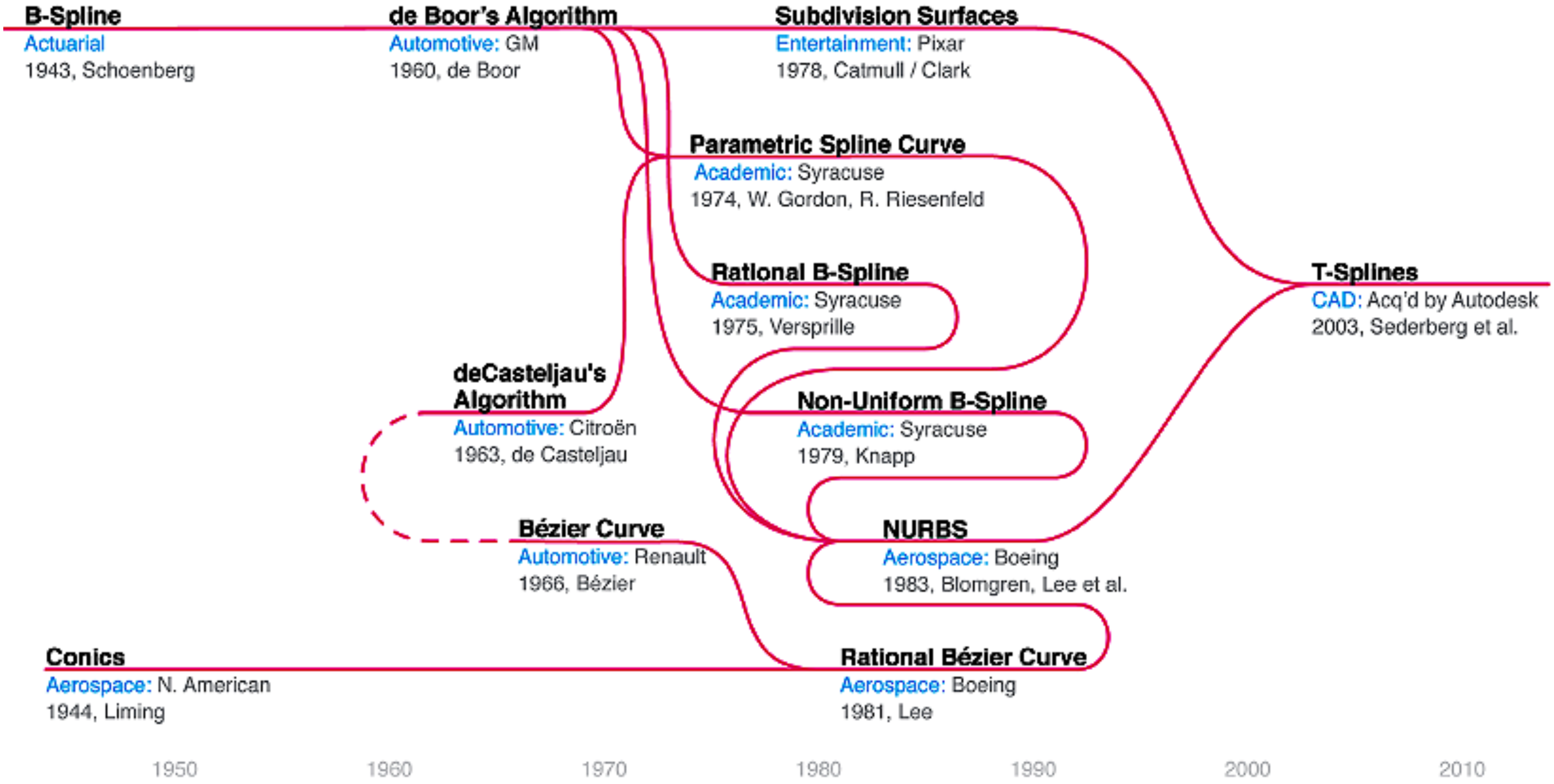
- Citroën’s DS 19 was introduced at the 1955 at the Paris Motor Show.
- Within the first 15 minutes of the show, 743 orders were received.

CAGD Precursor (new): Animations/VR/AR

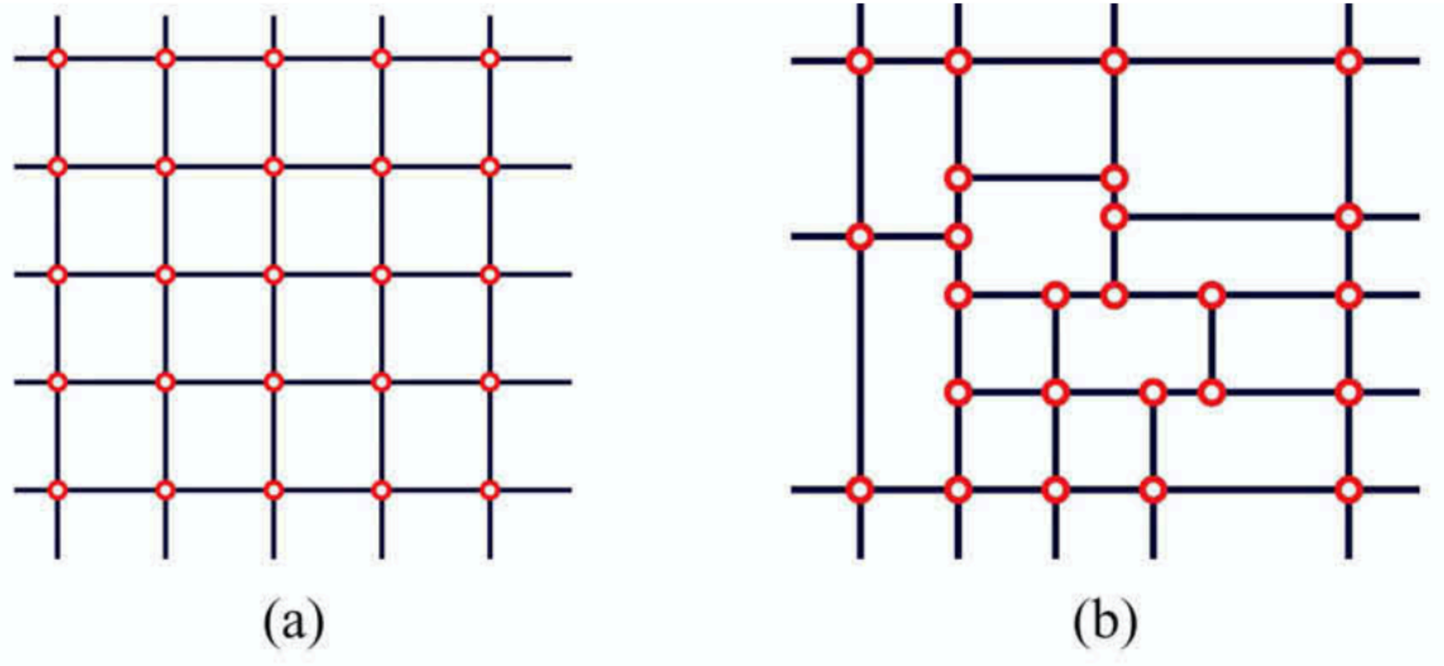


R&H
RHYTHM & HUES STUDIOS

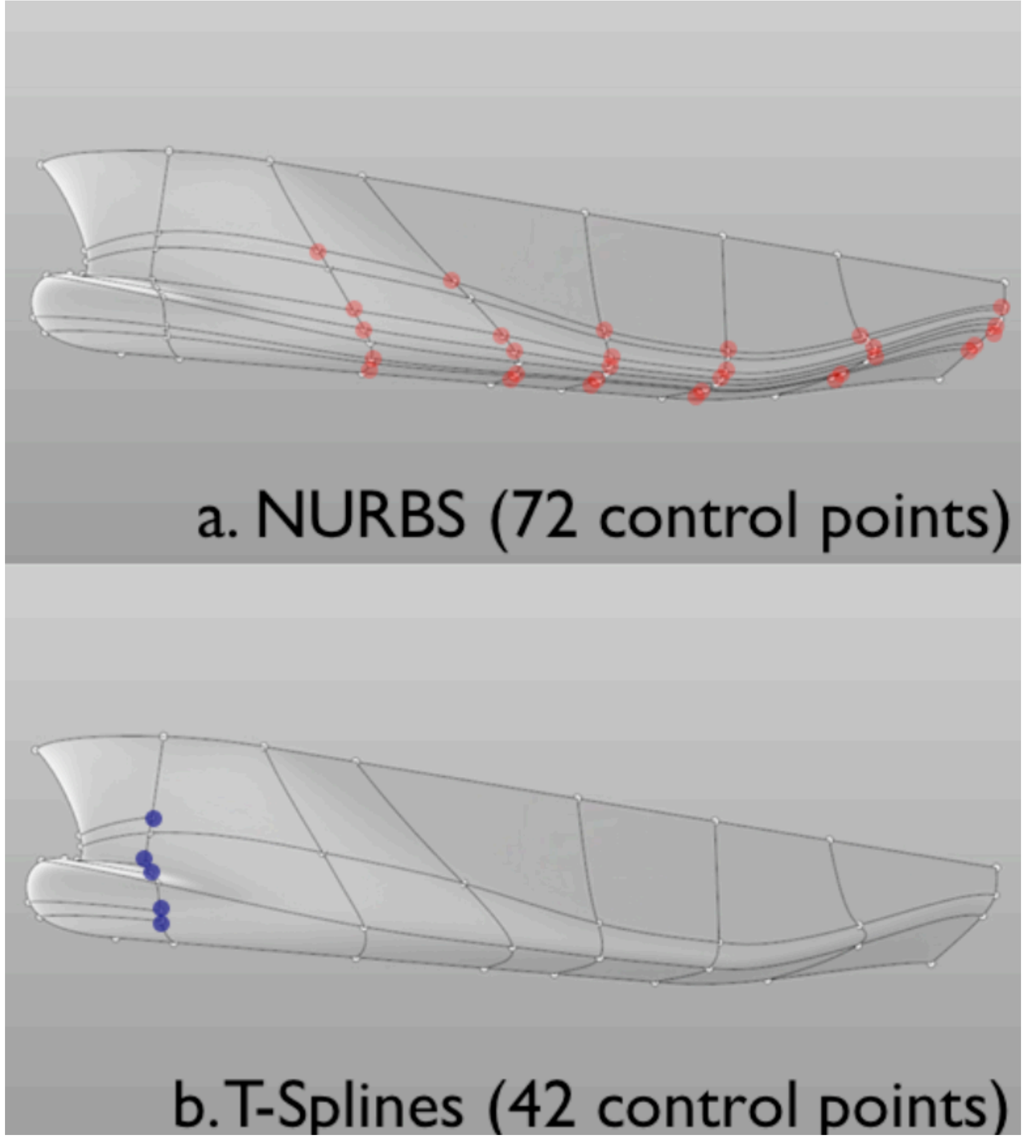
Timeline of flexible curves/surfaces



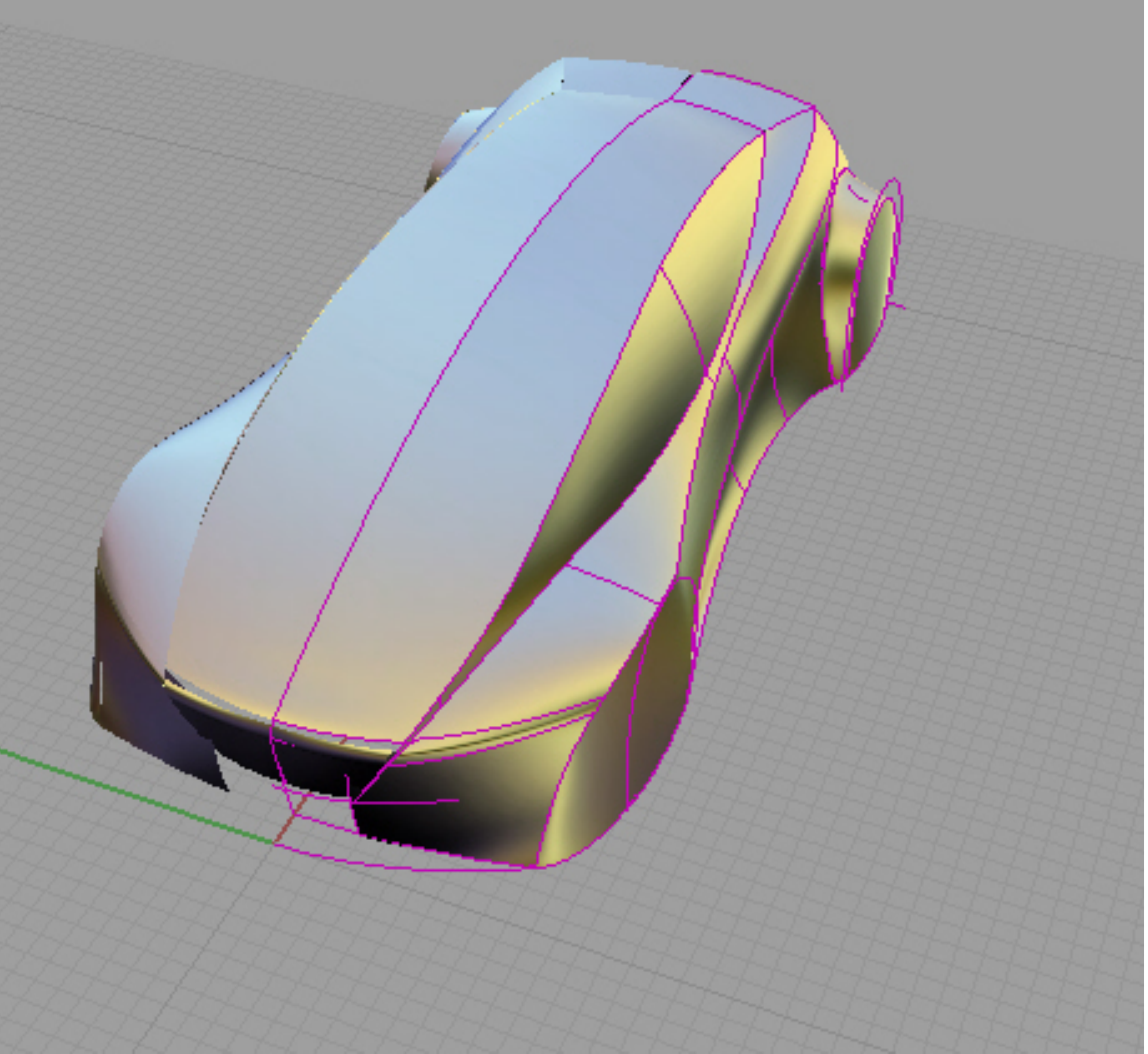
Interlude: T-Splines



Bazilevs et al, Isogeometric analysis using T-spline, Vol. 199, Issues 5–8, 2010, Pages 229-263



M.T Sederberg, T.W. Sederberg,T-Splines: A Technology for Marine Design with Minimal Control Points.



BEZIER CURVES!

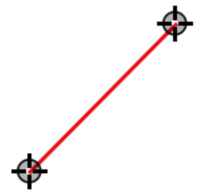
Mathematics of shapes

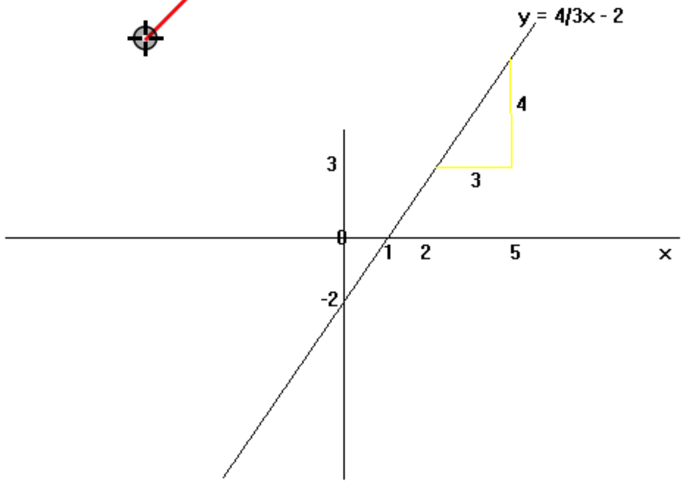
Curve representation, which is suitable?

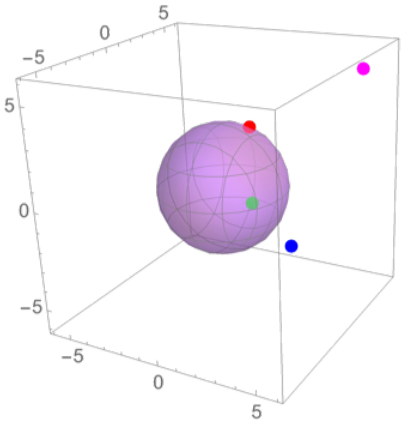
$$C(t) = (1-t)P_0 + (t)P_1 \quad , \quad t \in [0,1]$$

$$y = mx + c$$

$$ax + by = c$$





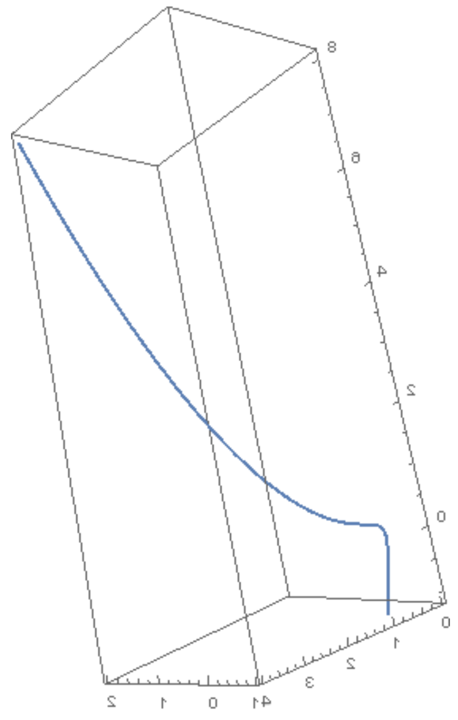


x^2 + y^2 + z^2 =10

Explicit & Implicit

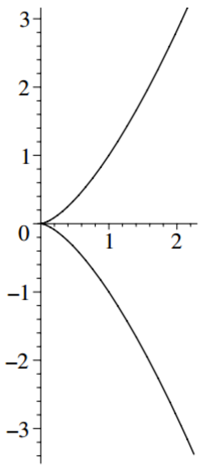
Example 5.3 (Non-parametric Implicit Curves)

1. Parabola: $y = x^2$, $x \in \mathbb{R}$.
2. Circular arc: $y = \sqrt{1 - x^2}$, $x \in [-1, 1]$.
3. Twisted space cubic: $y = x^2$, $z = x^3$, $x \in \mathbb{R}$.



Example 5.4 (Implicit Curves)

1. Unit radius circle: $x^2 + y^2 - 1 = 0$.
2. Cuspidal cubic: $y^2 - x^3 = 0$, see Figure 5.2.

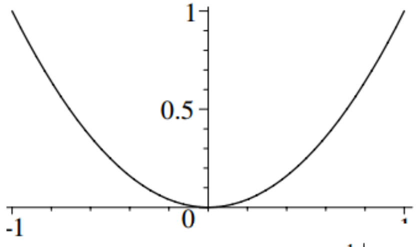


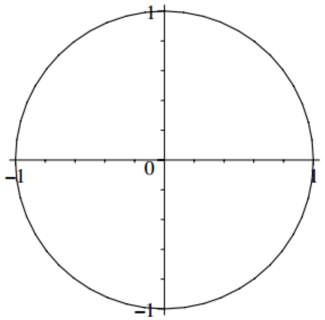
Cuspidal cubic $y^2 - x^3 = 0$

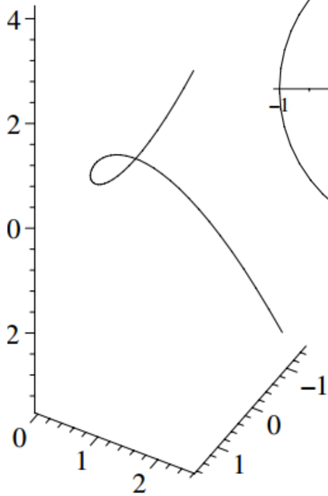
Parametric Curves

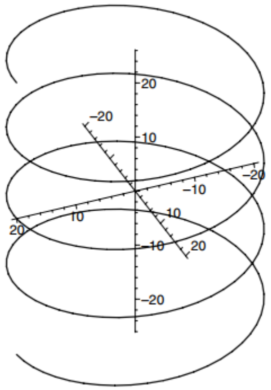
Example 5.2 (Parametric Curves)

1. Parabola: (t, t^2), for $t \in \mathbb{R}$, is a polynomial curve of degree 2. See Figure 5.1(a).
2. Quarter circle: $\left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right)$, for $t \in [0, 1]$, is a rational curve of degree 2.
3. Unit radius circle: $(\cos(t), \sin(t))$, for $t \in [0, 2\pi]$, see Figure 5.1(b).
4. Twisted space cubic: (t, t^2, t^3), for $t \in \mathbb{R}$ is a polynomial curve of degree 3. See Figure 5.1(c).
5. Helix: $(r\cos(t), r\sin(t), at)$, for $t \in \mathbb{R}$, $r > 0$, $a \neq 0$. See Figure 5.1(d).

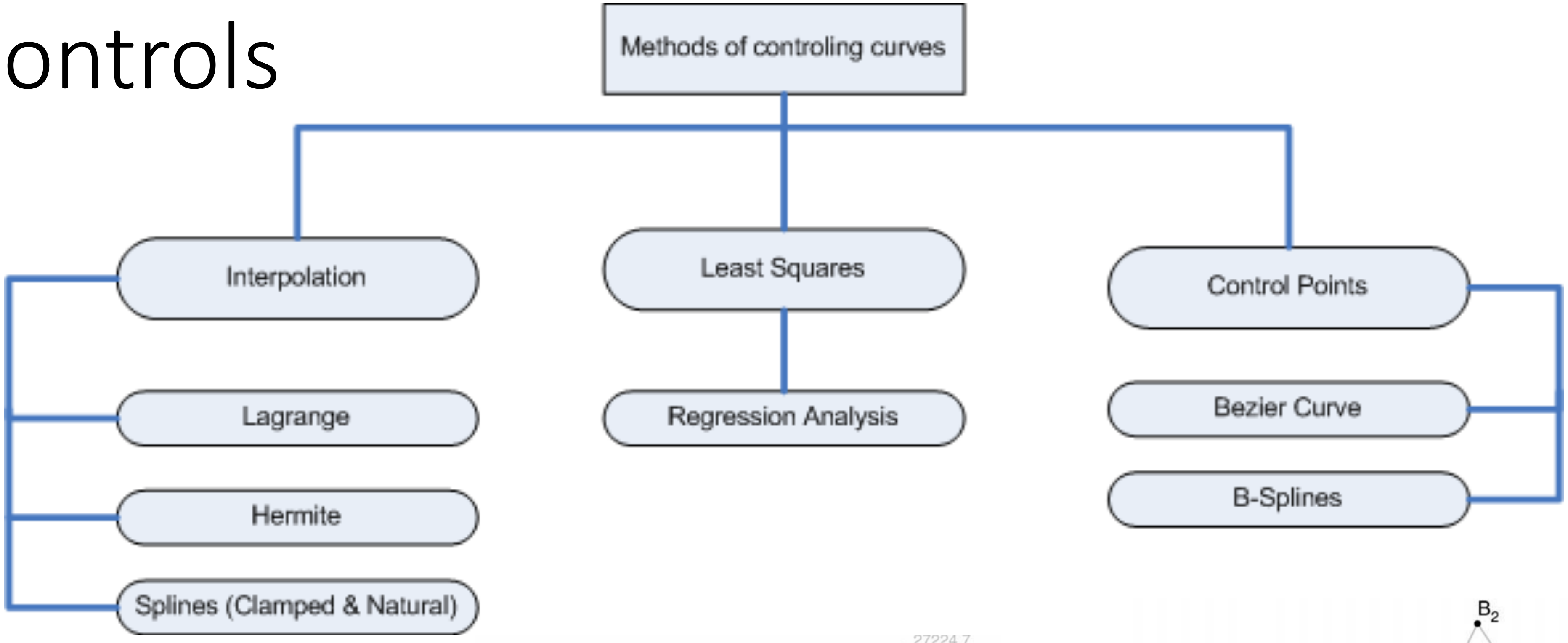


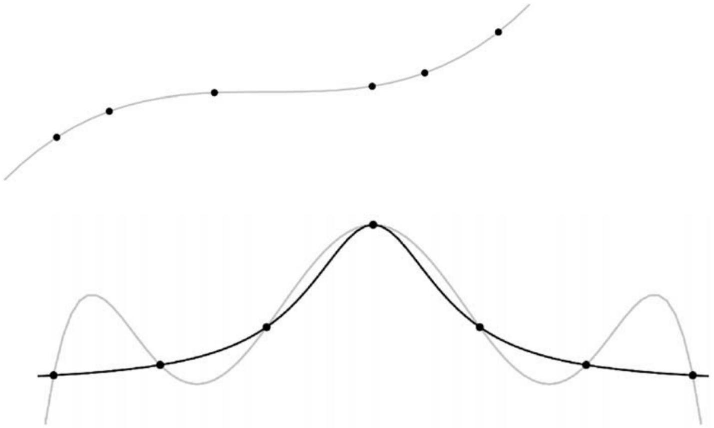


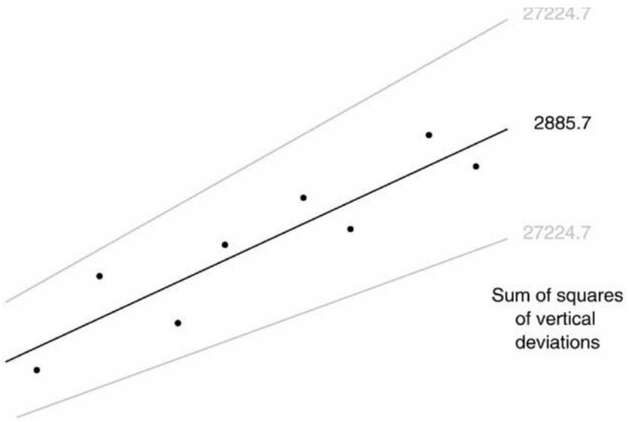


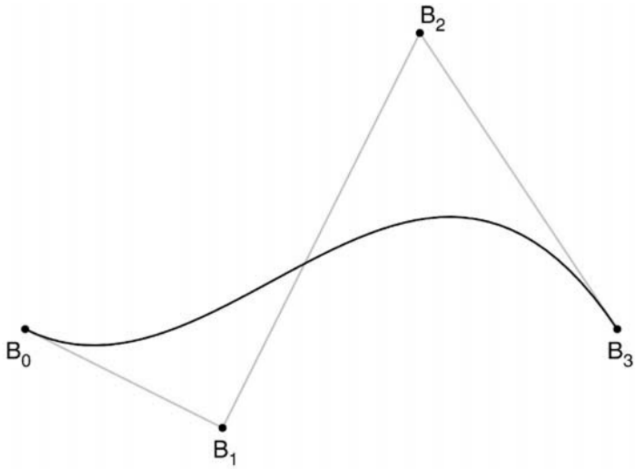


Curve controls









Classifications: curve design

- Mechanical Spline /ducks + strips
  - Flexible Spline
    - Conics
    - Hermite Spline
    - Bezier
      - R-Bezier
    - B- Spline
      - NURBS
        - T-Splines
  - Intrinsic curves

Bézier curve

- born 1930 in Besançon, France.
- In 1959 Citroën hired Paul de Faget de Casteljau, formulate a curve for design who just fresh from his PhD studies.

- Pierre Étienne Bézier (Sept.1, 1910 – Nov. 25, 1999)
- As an engineer at Renault, he became a leader in the transformation of design and manufacturing, through mathematics and computing tools, into computer-aided design and three-dimensional modeling.

de facto for CAD systems: B-Splines

- Isaac Jacob Schoenberg (April 21, 1903 – Feb. 21, 1990) was a Romanian-American mathematician

- Carl-Wilhelm Reinhold de Boor (born 3 Dec. 1937) is a German-American mathematician found a better way to evaluate B-Splines.
- ISI Highly Cited Author in Mathematics (at some point)

Ideal Curve for Design

We often wish to generate curves and surfaces

There are a number of (conflicting) goals

1. Interpolate (include) a set of arbitrary points
2. Easy and efficient to compute
3. Smooth (derivatives)
4. Local Control (vs. Global control)
5. Easy to deal with (easy create diff shapes)
6. Satisfy given geometric constraints

It is hard to get all of these…

Linear Bezier Curves = a line

A *linear Bézier curve* is a line segment joining two control points $\mathbf{b}_0(p_0, q_0)$ and $\mathbf{b}_1(p_1, q_1)$, and parametrized by

$$(x(t), y(t)) = (1-t)(p_0, q_0) + t(p_1, q_1), \quad \text{for } t \in [0,1] \, ,$$

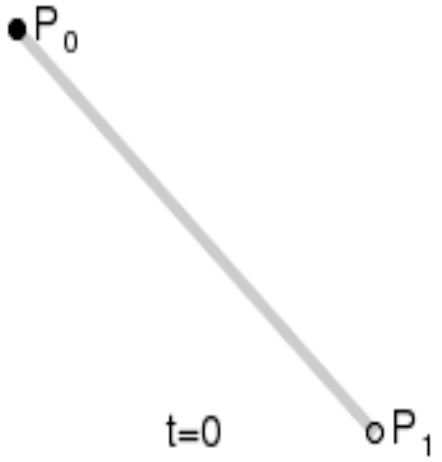
$$x(t) \; = \; (1-t)p_0 + tp_1, \text{ and } y(t) \; = \; (1-t)q_0 + tq_1.$$

- **Coordinate function = a polynomial**
- **Basis function**

$$\mathbf{B}(t) = (1-t)\mathbf{b}_0 + t\mathbf{b}_1 \; .$$

$$\mathbf{B}(0) \; = \; \mathbf{b}_0$$

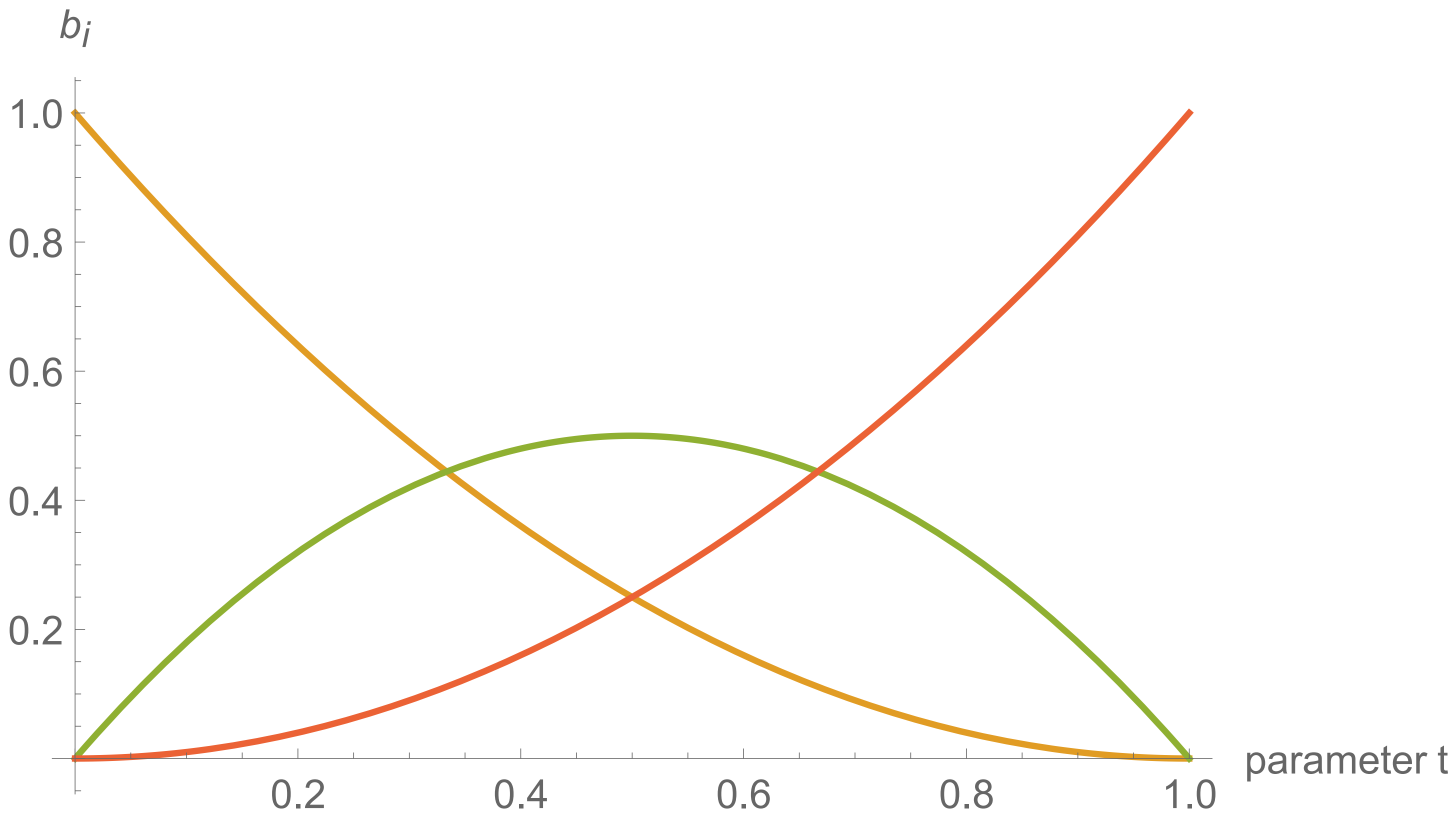
$$\mathbf{B}(1) \; = \; \mathbf{b}_1$$

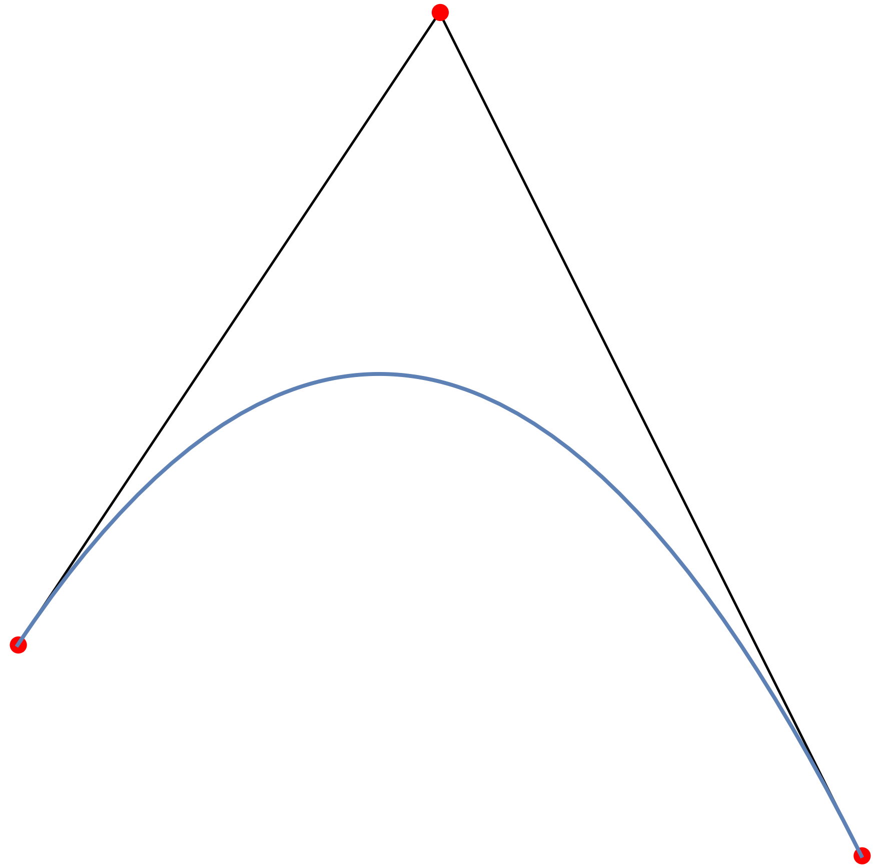


Beziers: characteristics

- Control point
- Control polygon

$$\mathbf{B}(t) = (1-t)^2(p_0, q_0) + 2(1-t)t(p_1, q_1) + t^2(p_2, q_2), \quad \text{for } t \in [0,1] .$$

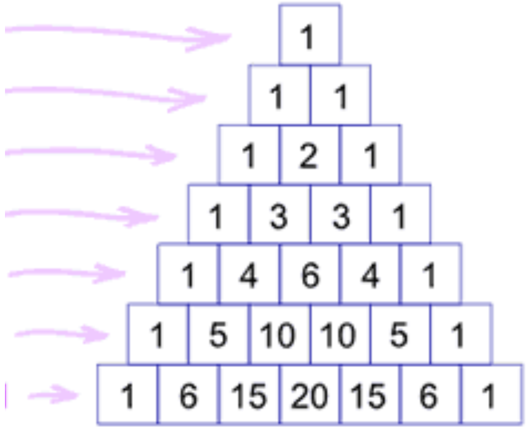




Beziers: general representation

Given $n+1$ control points $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_n$ the Bézier curve of degree n is defined to be

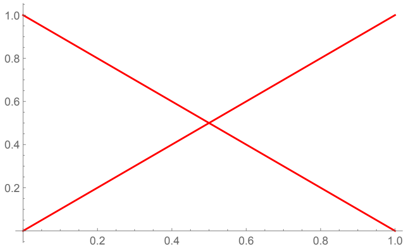
$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{b}_i B_{i,n}(t) \ ,$$

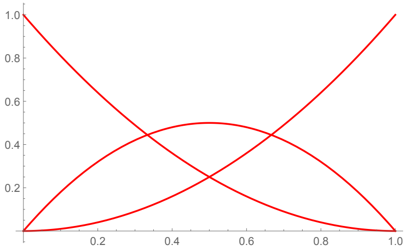


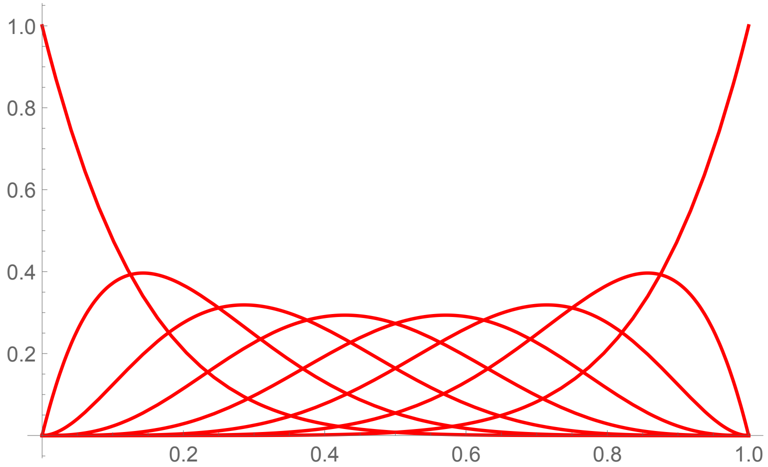
where

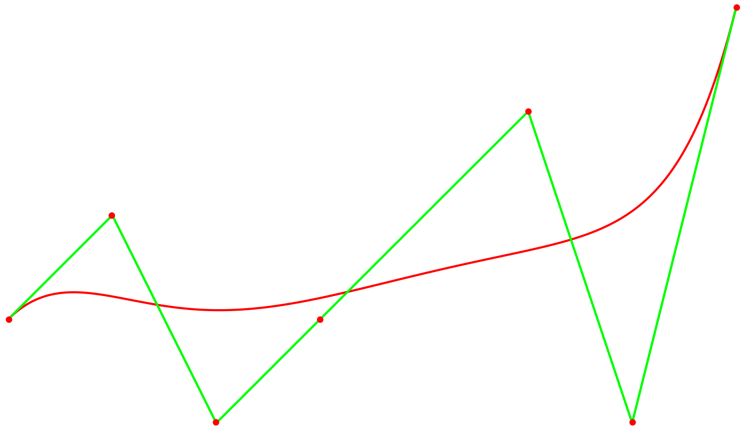
$$B_{i,n}(t) = \begin{cases} \frac{n!}{(n-i)!i!}(1-t)^{n-i}t^i, & \text{if } 0 \le i \le n \\ 0, & \text{otherwise} \end{cases}$$

are called the *Bernstein polynomials* or *Bernstein basis functions* of degree n.









Beziers: characteristics

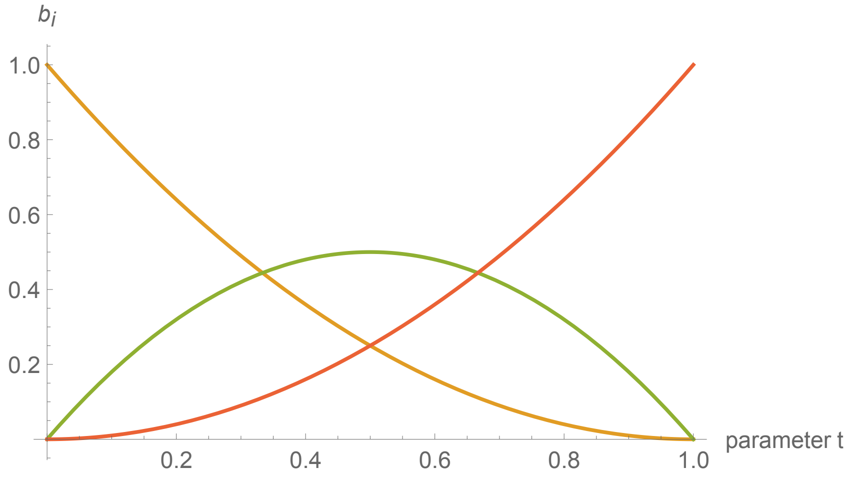
$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{b}_i B_{i,n}(t) \; ,$$

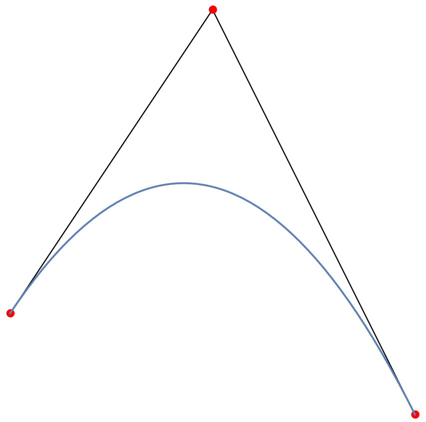
- **Partition of Unity** $\sum_{i=0}^{n} B_{i,n}(t) = 1, \quad t \in [0,1] \; .$
- **Positivity** $B_{i,n}(t) \geq 0, \quad t \in [0,1] \; .$
- **Symmetry** $B_{n-i,n}(t) = B_{i,n}(1-t) \; , \text{ for } i = 0, \ldots, n \; .$
- **Recursion** $B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t)$

1. Convex Hull Property
2. Invariance under transformation

3. Symmetrical control polygon results symmetrical curve

4. de Casteljau algorithm : easy computation!





5. Variation diminishing property
6. Endpoint Interpolation.
7. Endpoint Tangent Property.

Beziers: rendering using de Casteljau Algo.

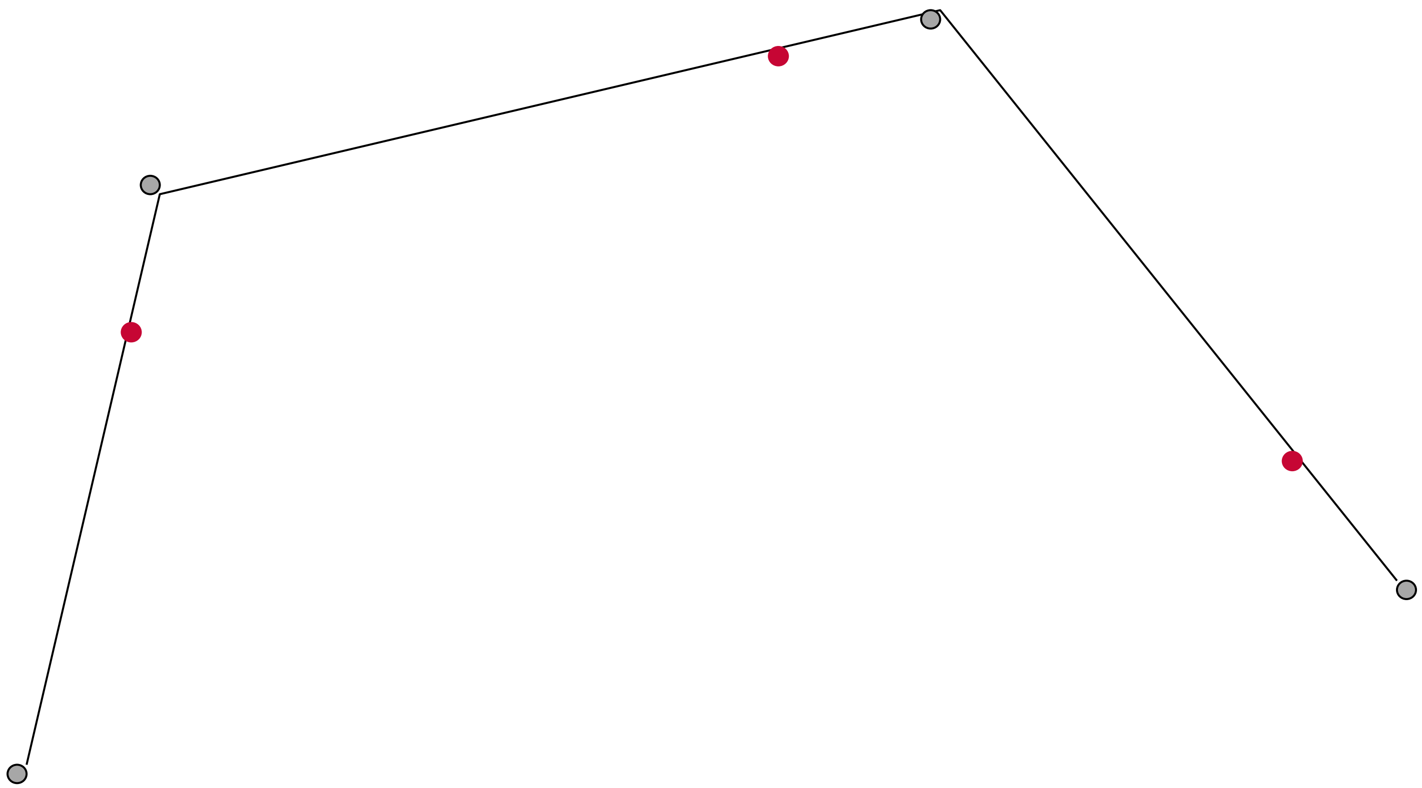
$$\begin{cases} \mathbf{b}_i^0 = \mathbf{b}_i, \\ \mathbf{b}_i^j = (1-t)\mathbf{b}_i^{j-1} + t\mathbf{b}_{i+1}^{j-1}, \end{cases}$$

$$\begin{array}{llll} \mathbf{b}_0^0 & \mathbf{b}_1^0 & \mathbf{b}_2^0 & \mathbf{b}_3^0 \\ \mathbf{b}_0^1 & \mathbf{b}_1^1 & \mathbf{b}_2^1 & \\ \mathbf{b}_0^2 & \mathbf{b}_1^2 & & \\ \mathbf{b}_0^3 & & & \end{array}$$

- Can compute any point on the curve in a few iterations
- No polynomials involved
- Repeated linear interpolation

de Casteljau algorithm

$$
\begin{array}{cccc}
\mathbf{b}_0^0 & \mathbf{b}_1^0 & \mathbf{b}_2^0 & \mathbf{b}_3^0 \\
\mathbf{b}_0^1 & \mathbf{b}_1^1 & \mathbf{b}_2^1 & \\
\mathbf{b}_0^2 & \mathbf{b}_1^2 & & \\
\mathbf{b}_0^3 & & &
\end{array}
$$



Degree 3, t=0.75

Rational Bezier: more shape control

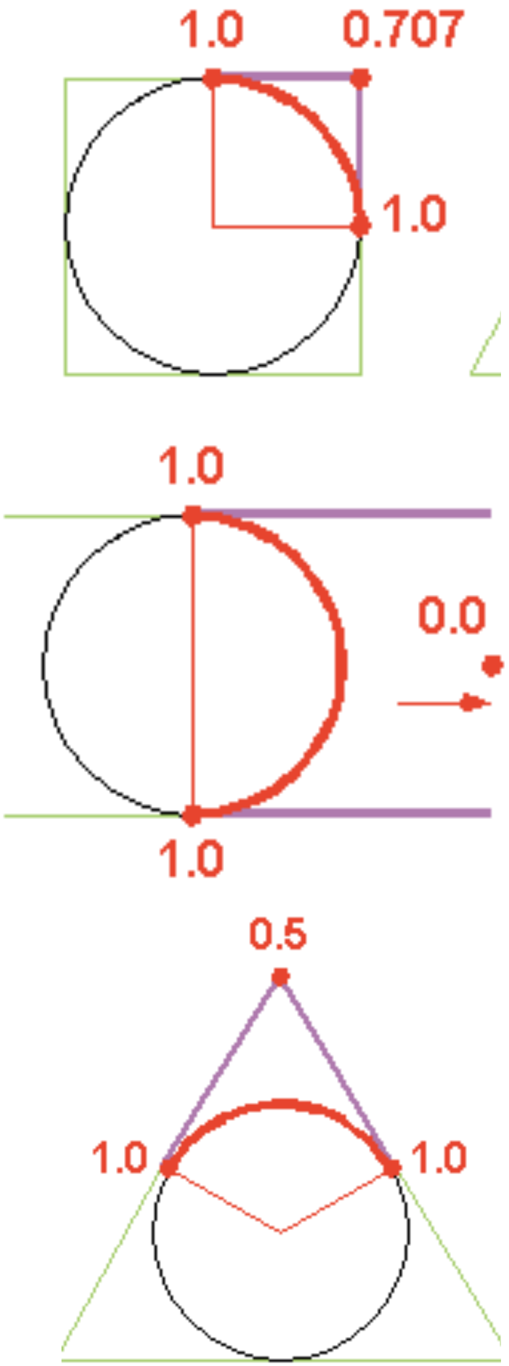
Definition 7.19

A *rational Bézier curve* of degree n with control points $\mathbf{b}_0, \ldots, \mathbf{b}_n$ and corresponding scalar *weights* w_i is defined to be

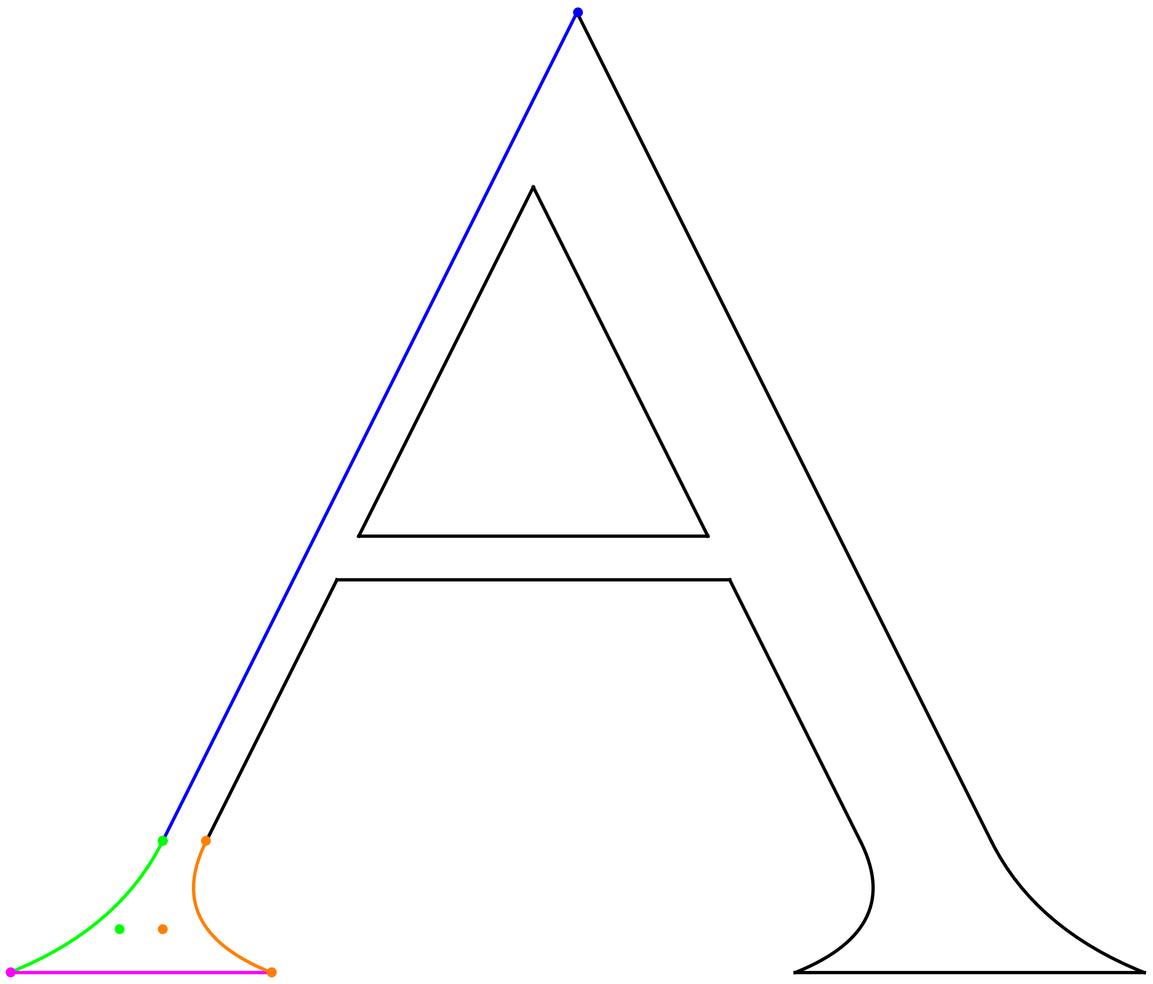
$$\mathbf{B}(t) = \frac{\sum_{i=0}^{n} w_i \mathbf{b}_i B_{i,n}(t)}{\sum_{i=0}^{n} w_i B_{i,n}(t)} , \quad t \in [0, 1] ,$$

with the understanding that if $w_i = 0$, then $w_i \mathbf{b}_i$ is to be replaced by $\mathbf{b}_i$. It is assumed that not all the weights are zero. When $\mathbf{b}_i \in \mathbb{R}^2$ $(i = 0, \ldots, n)$ then the curve is planar, and when $\mathbf{b}_i \in \mathbb{R}^3$ the curve is spatial. Note that the term *integral* Bézier curve is used to describe non-rational Bézier curves.

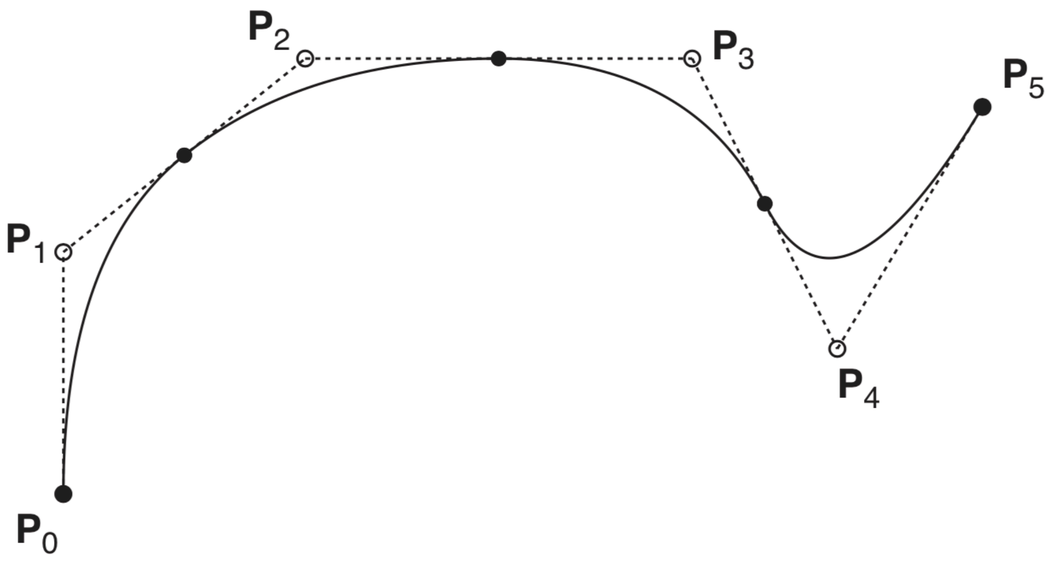
Since $\sum_{i=0}^{n} B_{i,n}(t) = 1$, by the partition of unity property, integral Bézier curves are obtained whenever $w_0 = \cdots = w_n$.



Bezier Curves: applications

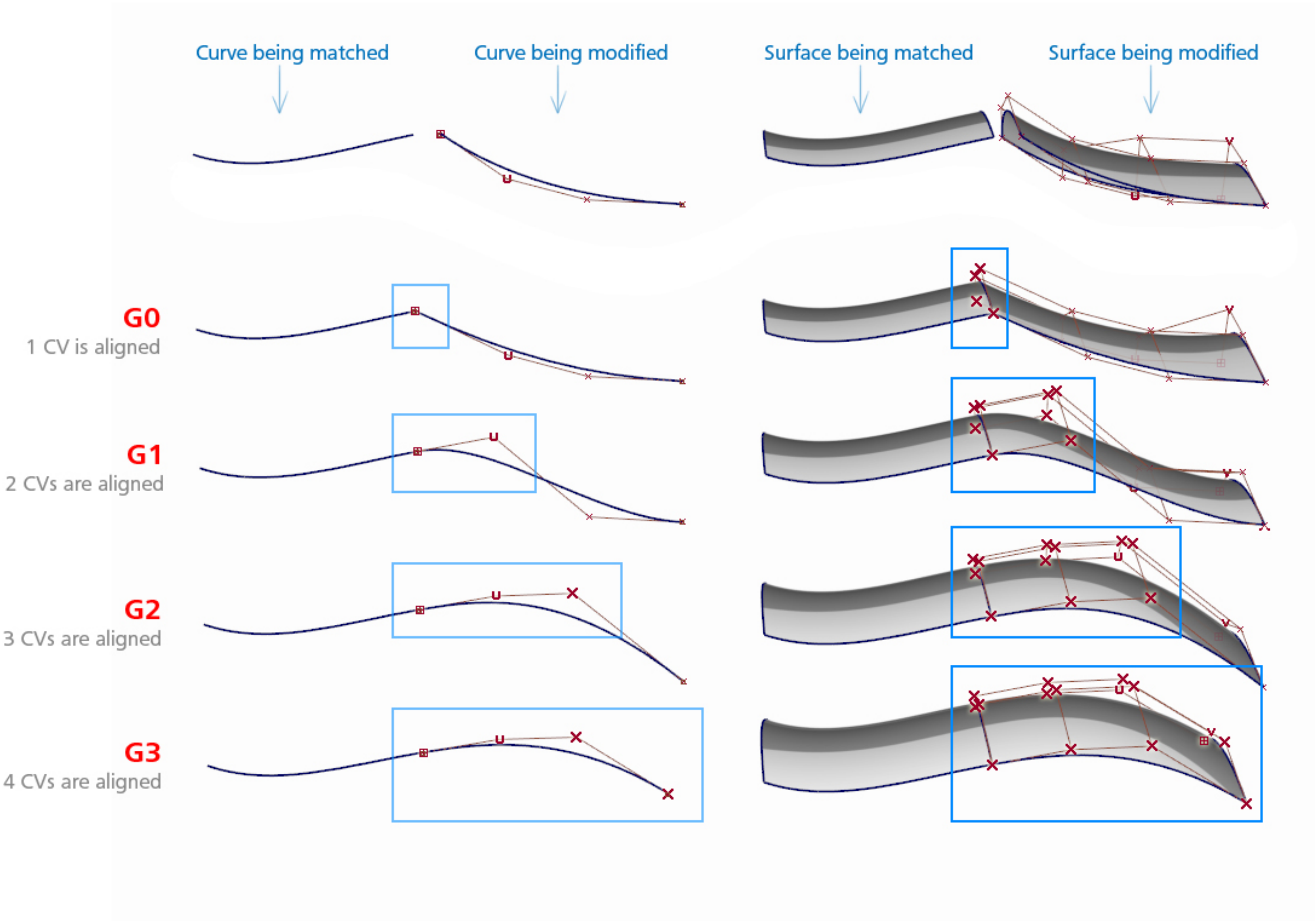


outline of a glyph



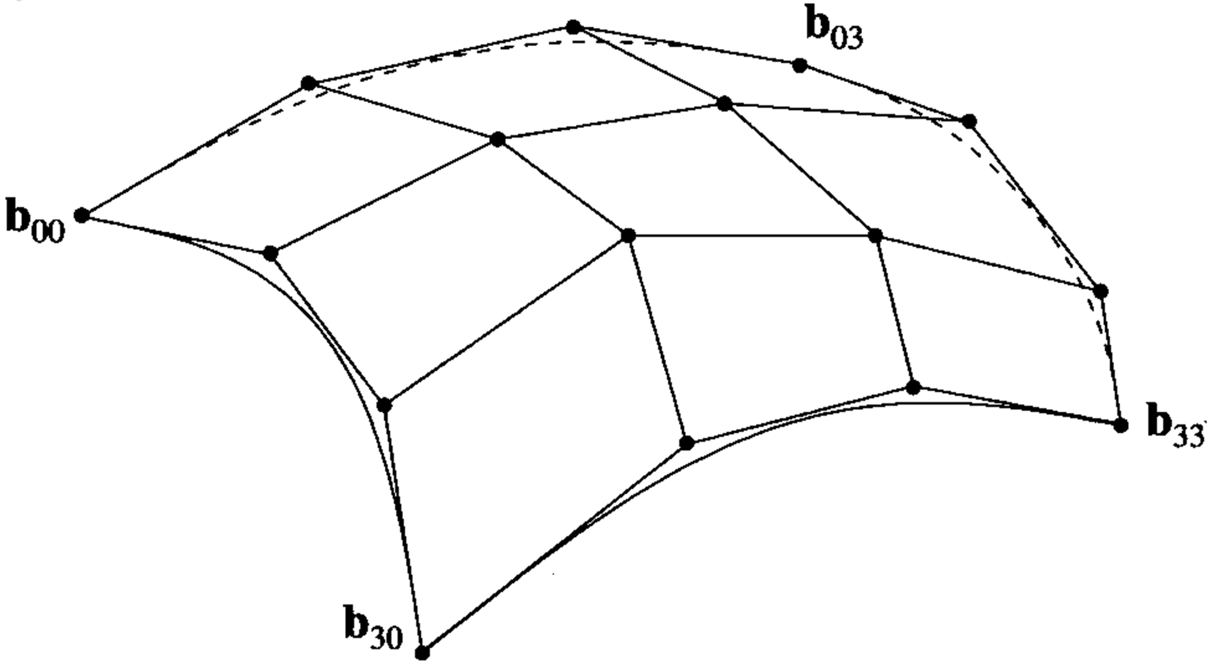
Path design

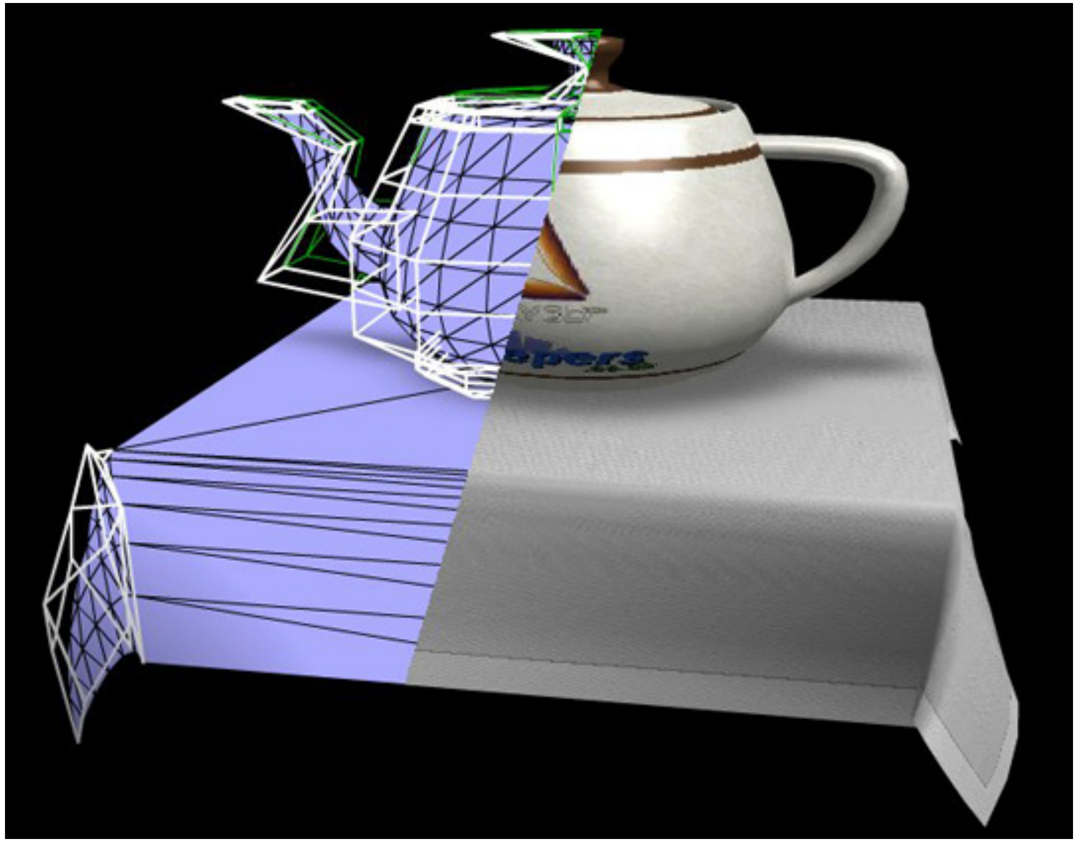
Continuity and Bezier control points



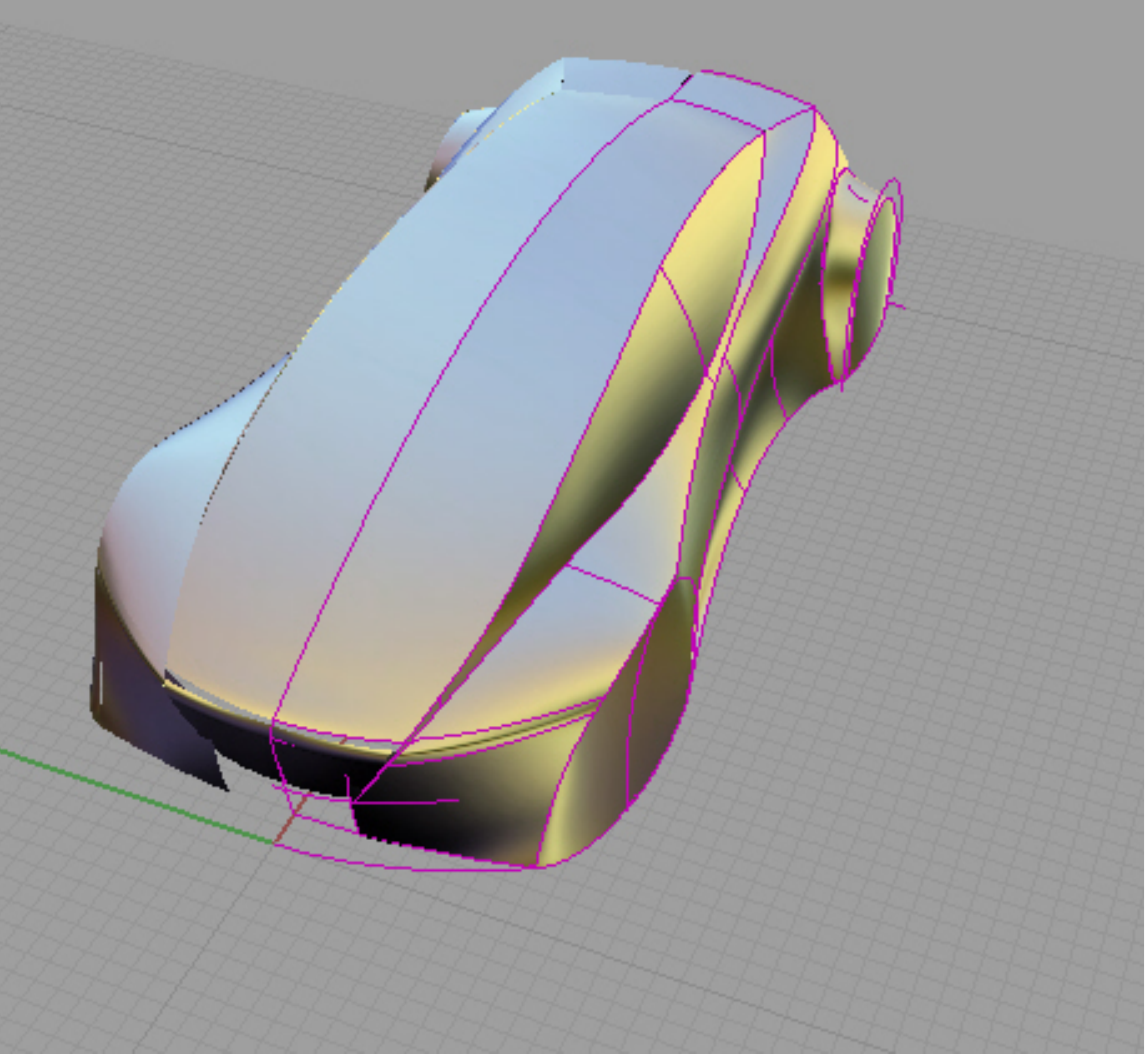
Sneak peek: Bezier Surface.

Tensor product





$$\mathbf{S}(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{p} \mathbf{p}_{i,j} B_{i,n}(s) B_{j,p}(t), \text{ for } (s,t) \in [0,1] \times [0,1] \,.$$

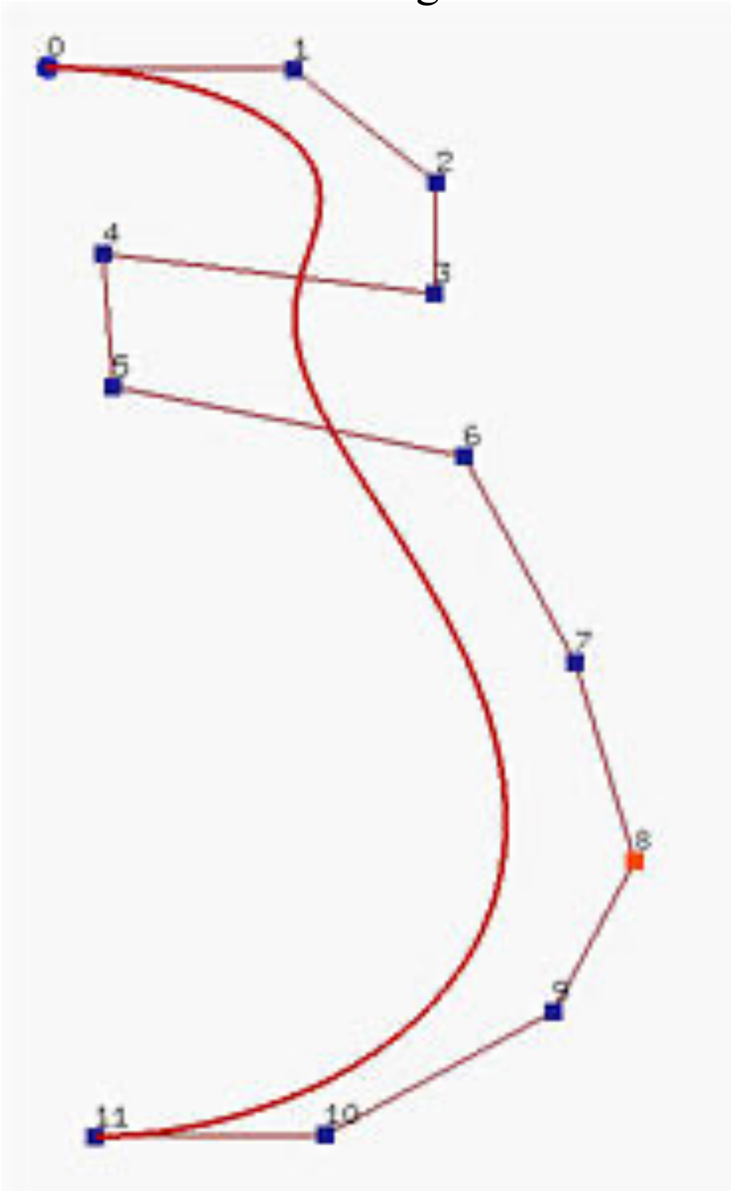


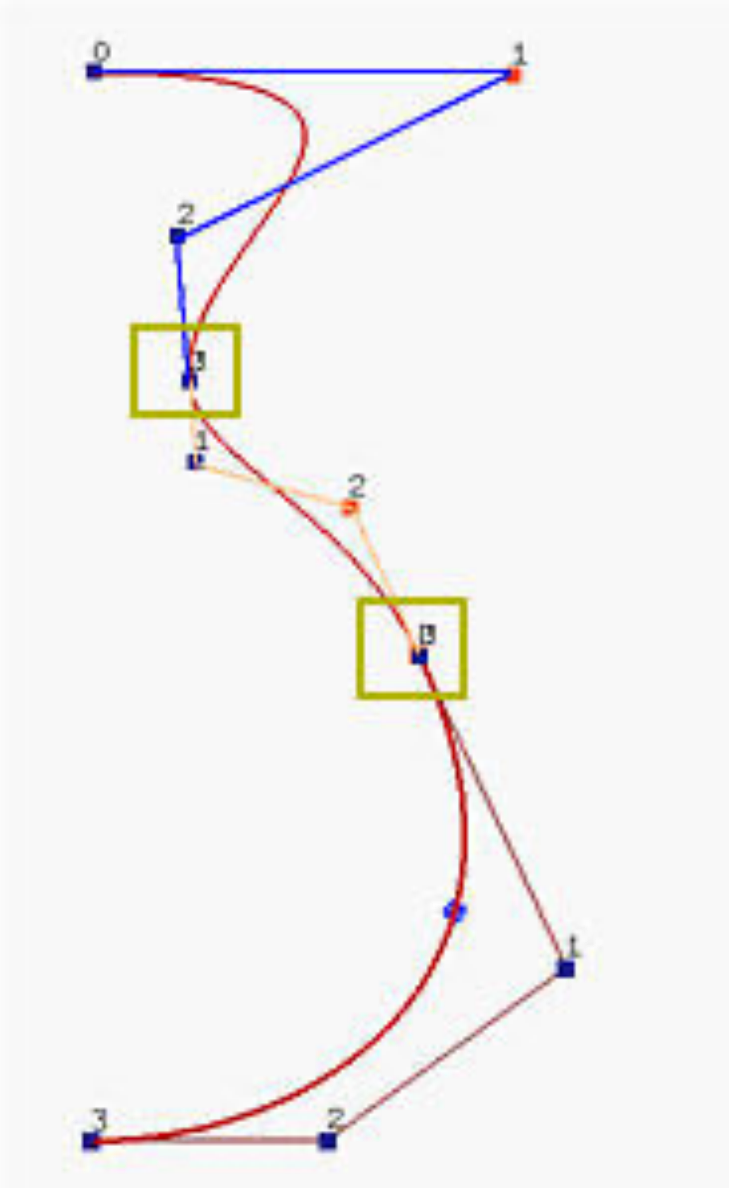
B-SPLINES

Mathematics of shapes

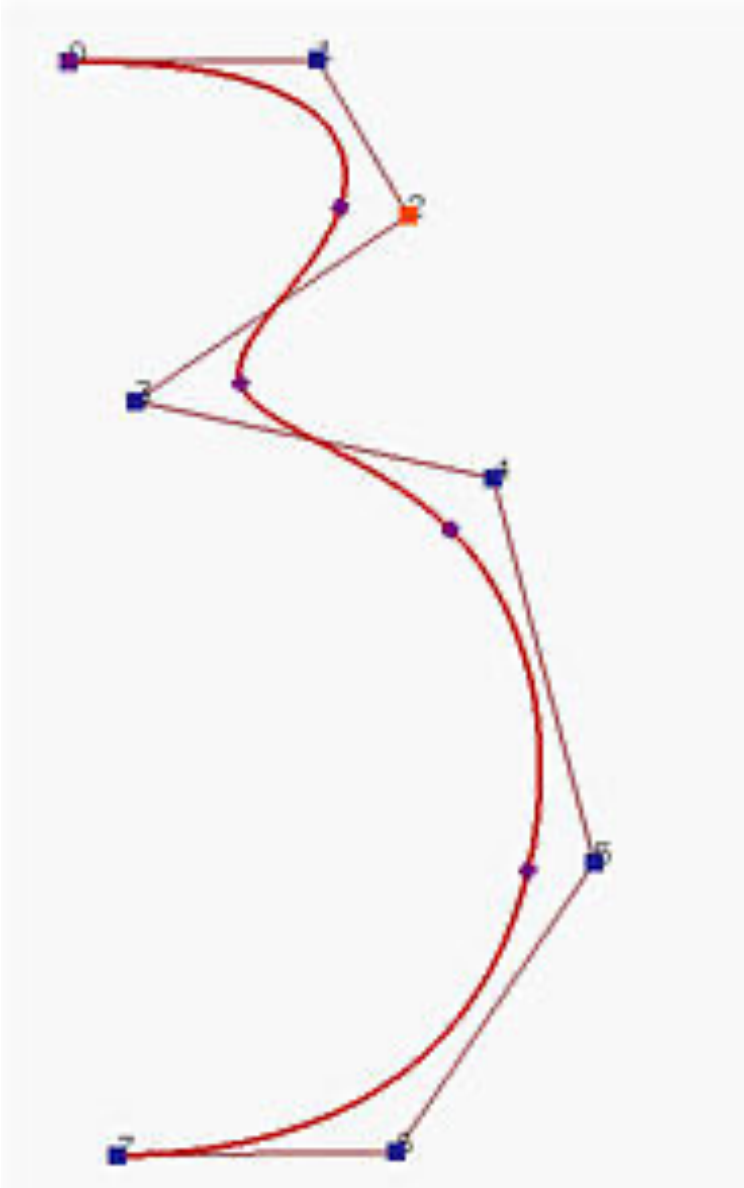
Motivation: B-Spline Basis

Bézier curve of degree 11





three Bézier curve segments of degree 3



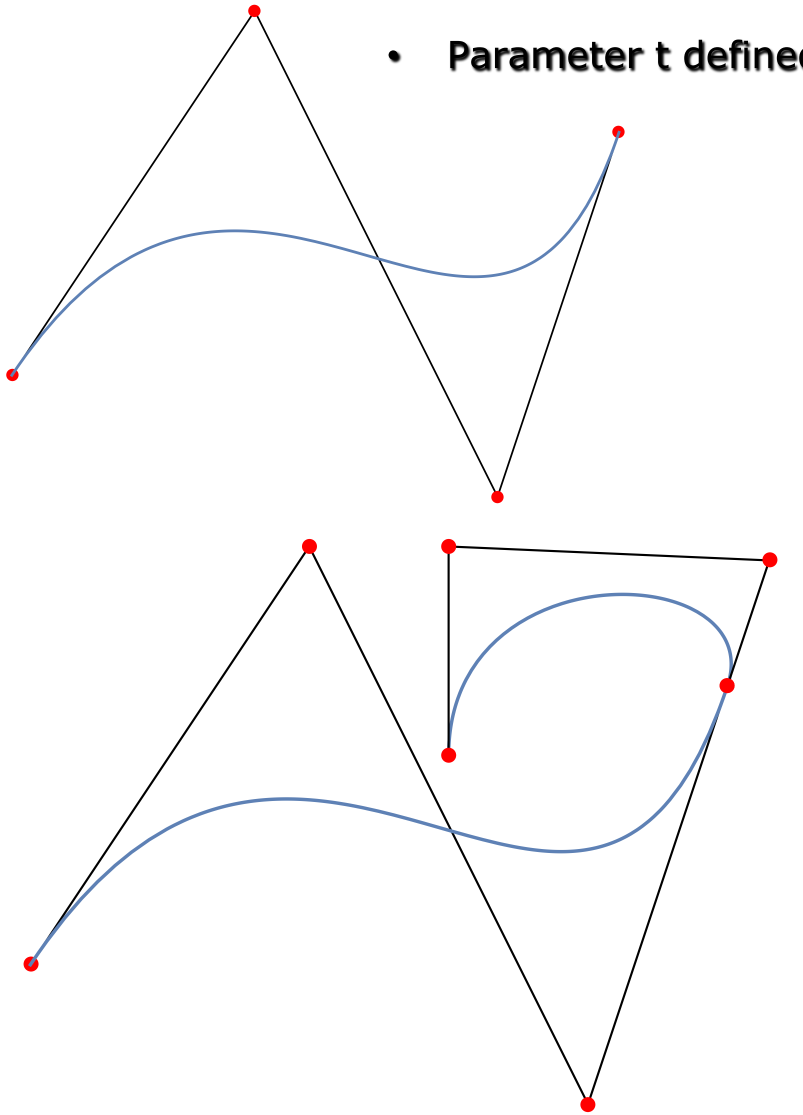
B-spline curve of degree 3 (8 control points)

B-Spline Curves
(Two Advantages)

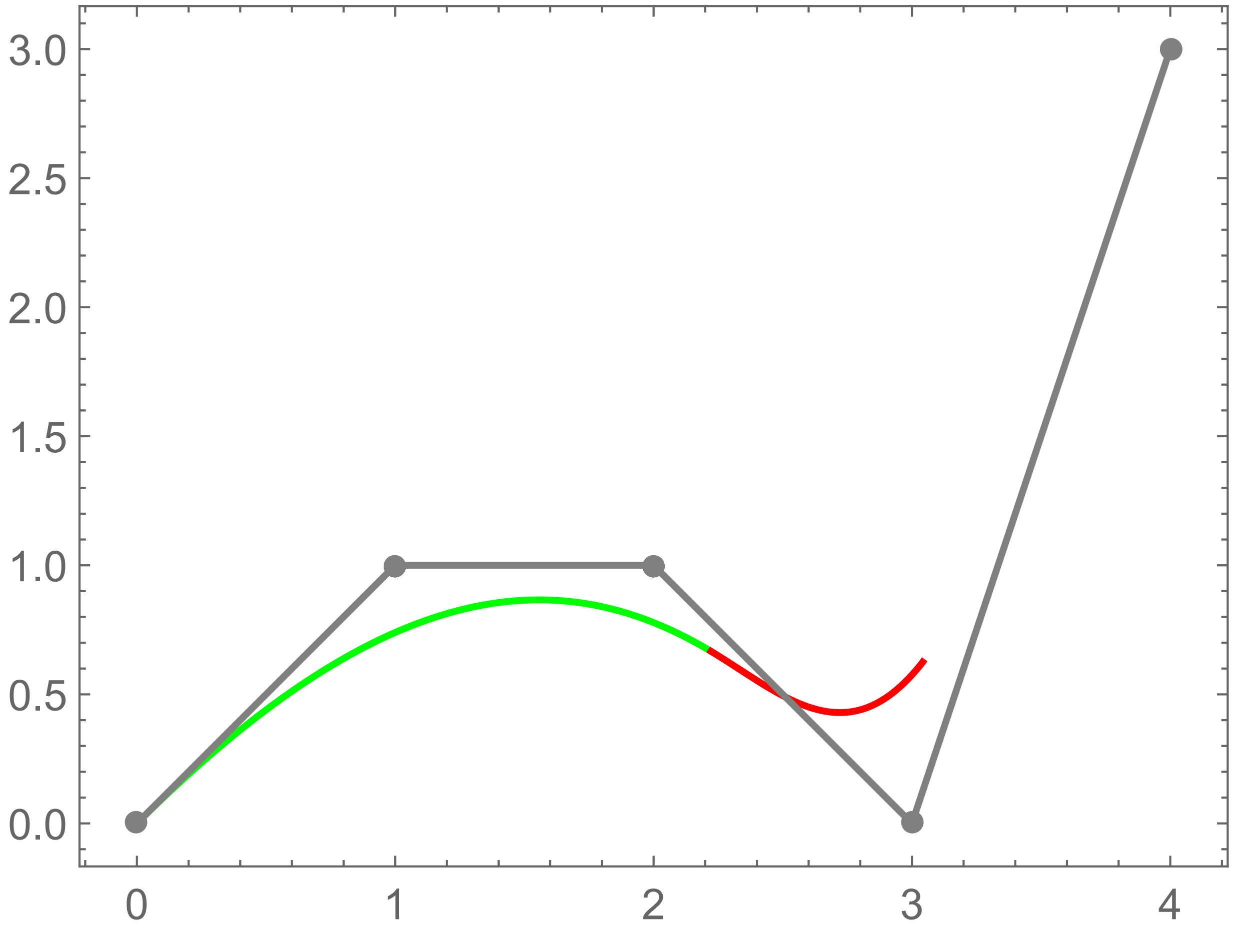
1. The degree of a B-spline polynomial can be set independently of the number of control points.

2. B-splines allow local control over the shape of a spline curve (or surface)

Bezier Vs B-spline: Something special about BSpline

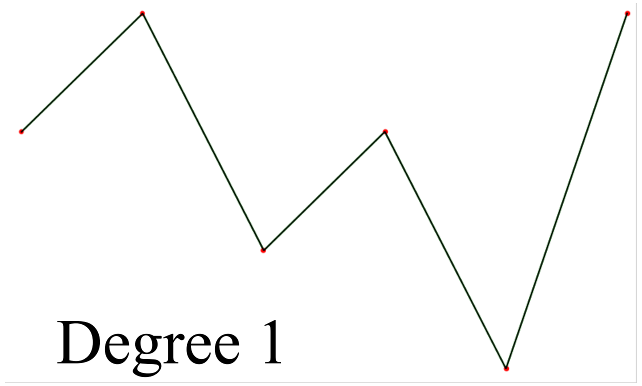


- Parameter t defined in [0,1]

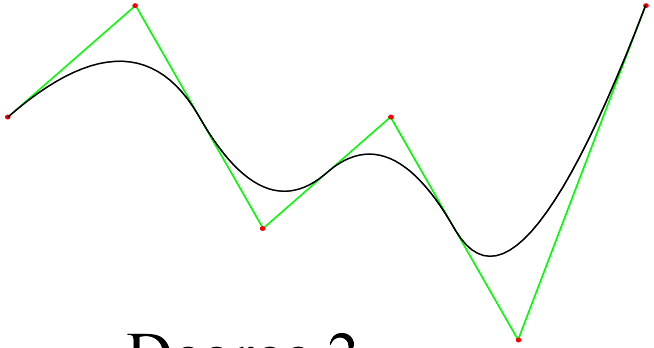


- Many parameters: knot vectors

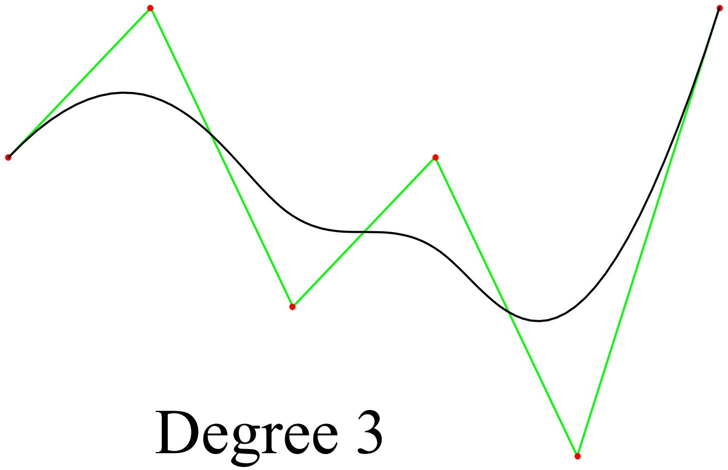
B-Spline Curves: degree increment



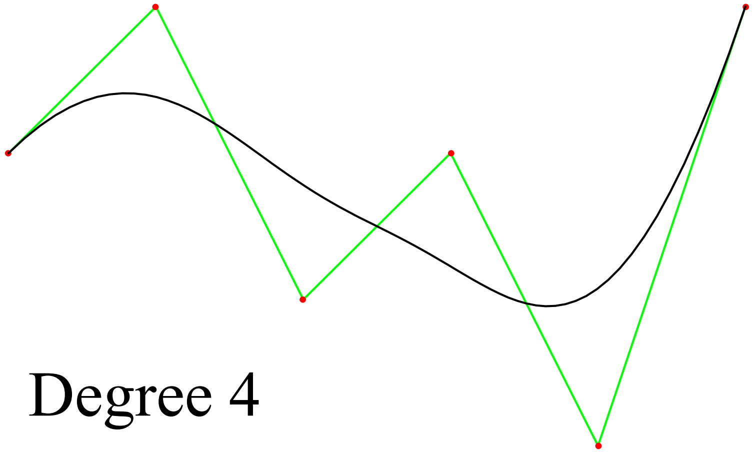
Degree 1



Degree 2



Degree 3



Degree 4

B-Spline

The *B-spline curve* of degree d (or order $d+1$) with control points $\mathbf{b}_0, ..., \mathbf{b}_n$ and knots $t_0, ..., t_m$ is defined on the interval $[a, b] = [t_d, t_{m-d}]$ by

$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{b}_i N_{i,d}(t) \,,$$

$$\boxed{m = n + d + 1 \,.}$$

where $N_{i,d}(t)$ are the B-spline basis functions of degree d. To distinguish B-spline curves from their rational form (which will be introduced in Section 8.2) they are often referred to as *integral* B-splines.

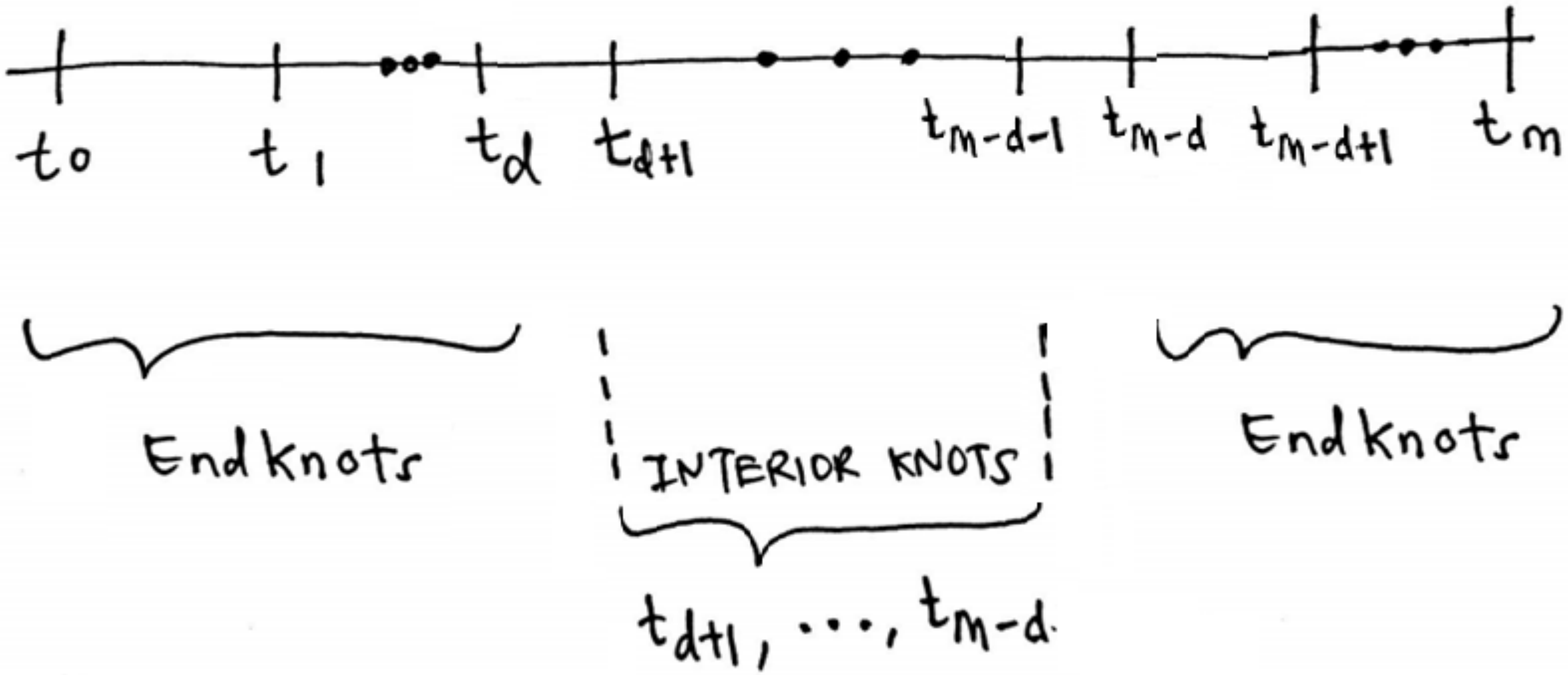
$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1, & \text{if } t \in [t_i, t_{i+1}) \\ 0, & \text{otherwise} \end{cases} \,, \\ N_{i,d}(t) &= \frac{t - t_i}{t_{i+d} - t_i} N_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(t) \,, \end{aligned}$$

Evaluation?

- **de Boor Algorithm**

'many parameters' = Knots vectors

degree of $\underline{\underline{d}}$ or order of $\underline{\underline{d+1}}$



B-Spline defined in $t \in [t_d, t_{m-d}]$

How make it interpolate endpoints?

A knot vector can have repeated knot values. The number of times a knot value occurs is called the *multiplicity* of the knot. Define a new sequence $u_0, \ldots, u_r$ ($u_0 < \cdots < u_r$), called the *breakpoints*, consisting of the distinct values of the interior knots. Then the B-spline is the union of the polynomial curve segments $\mathbf{B}_i(t)$ of degree d, $t \in [u_i, u_{i+1})$.

In general, B-spline curves do not interpolate the first and last control points $\mathbf{b}_0$ and $\mathbf{b}_n$. For curves of degree d, endpoint interpolation and an endpoint tangent condition are obtained by *open B-splines* for which the end knots satisfy $t_0 = t_1 = \ldots = t_d$ and $t_{m-d} = t_{m-d+1} = \ldots = t_m$.

B-Spline: Example 1

a. Find the number of knots needed for a B-Spline curve which has degree = 2 and 3 control points (p_0 , p_1 & $p_{n=2}$).

- d=2, Control points, n=2,
- → $m= n + d + 1 = 5$: $t_0, t_1, t_2, t_3, t_4, t_5$.

b. Find the knot values such a way that it interpolates endpoints.

- $t_0=0, t_1=0, t_2=0, t_3=1, t_4=1, t_5= 1$

c. Find nu of segments within this B-spline.

- $[t_d, t_{m-d}] = [t_2, t_{5-2=3}] = [t_2, t_3]= [0, 1]$

d. Finally, find the equation of B-Spline curve in polynomial form.

Non-periodic (open) uniform B-Spline

Answer d:

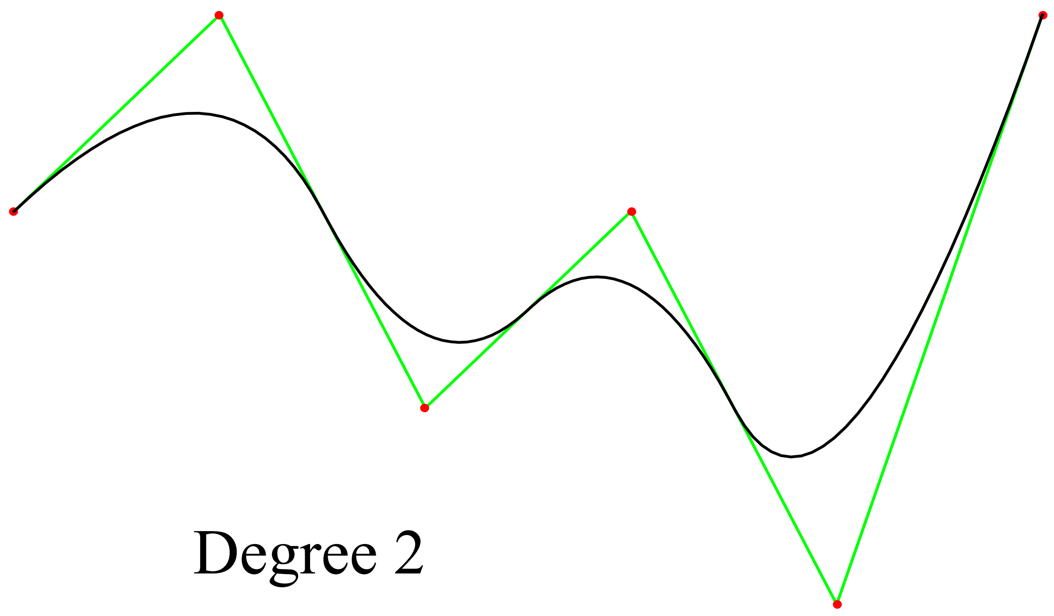
1. To obtain the basis function,

$$B(t) = \sum_{i=0}^{n} N_{i,d}(t)p_i$$

- $= \sum_{i=0}^{2} N_{i,2}(t)p_i$
- $= N_{0,2}(t)p_0 + N_{1,2}(t)p_1 + N_{2,2}(t)p_2$

2. Firstly, find the $N_{i,d}(t)$ using the knot value that shown above, start from d =0 to d=2.

Non-periodic (open) uniform B-Spline

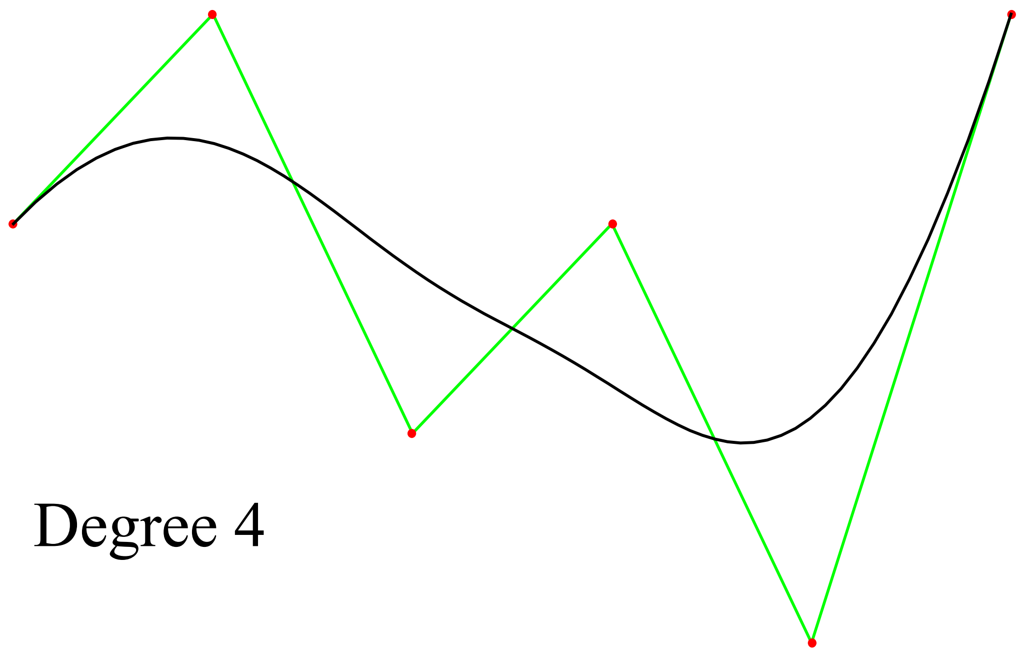


Degree 2

- $n=?$, $d=2$, $m=n+d+1=?$
- $[t_d, t_{m-d}] =$
- Number of segments?
- Each set of control points?

Non-periodic (open) uniform B-Spline

- $n=?$, $d=4$, $m=n+d+1=?$
- $[t_d, t_{m-d}] =$
- Number of segments?
- Each set of control points?



Degree 4

B-spline's characteristics

Theorem 8.6

A B-spline curve $\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{b}_i N_{i,d}(t)$ of degree d defined on the knot vector $t_0, \ldots, t_m$ satisfies the following properties.

Local Control: Each segment is determined by $d+1$ control points. If $t \in [t_r, t_{r+1})$ $(d \leq r \leq m-d-1)$, then

$$\mathbf{B}(t) = \sum_{i=r-d}^{r} \mathbf{b}_i N_{i,d}(t) \ .$$

Thus to evaluate $\mathbf{B}(t)$ it is sufficient to evaluate $N_{r-d,d}(t), \ldots, N_{r,d}(t)$.

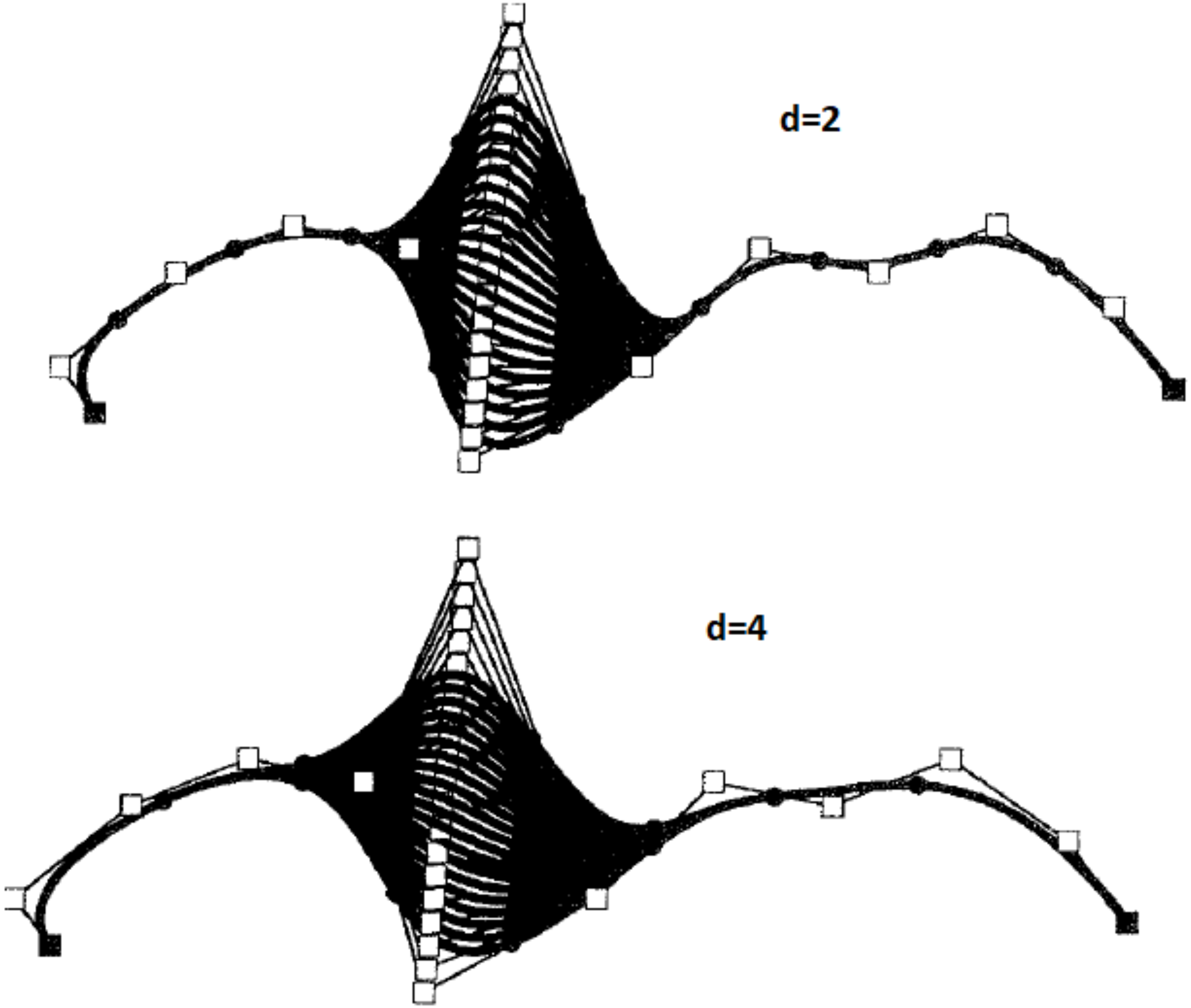
Convex Hull: If $t \in [t_r, t_{r+1})$ $(d \leq r \leq m-d-1)$, then

$$\mathbf{B}(t) \in \mathrm{CH}\{\mathbf{b}_{r-d}, \ldots, \mathbf{b}_r\} \ .$$

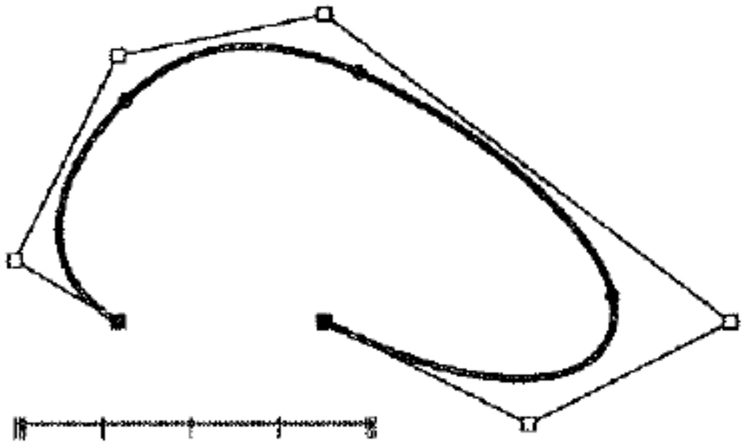
Continuity: If p_i is the multiplicity of the breakpoint $t = u_i$, then $\mathbf{B}(t)$ is C^{d-p_i} (or greater) at $t = u_i$, and C^∞ elsewhere.

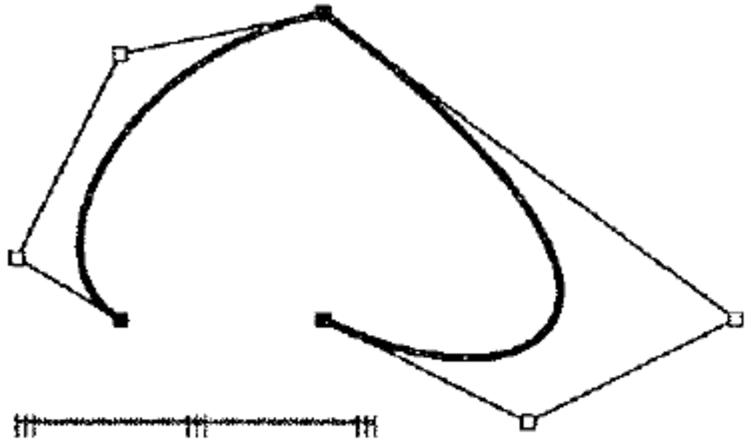
Invariance under Affine Transformations: Let T be an affine transformation. Then $\mathsf{T}\left(\sum_{i=0}^{n} \mathbf{b}_i N_{i,d}(t)\right) = \sum_{i=0}^{n} \mathsf{T}(\mathbf{b}_i) N_{i,d}(t)$.

Local control

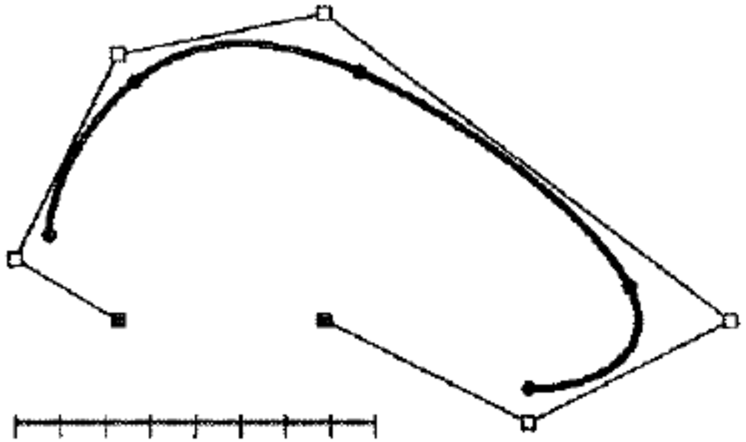


Continuity

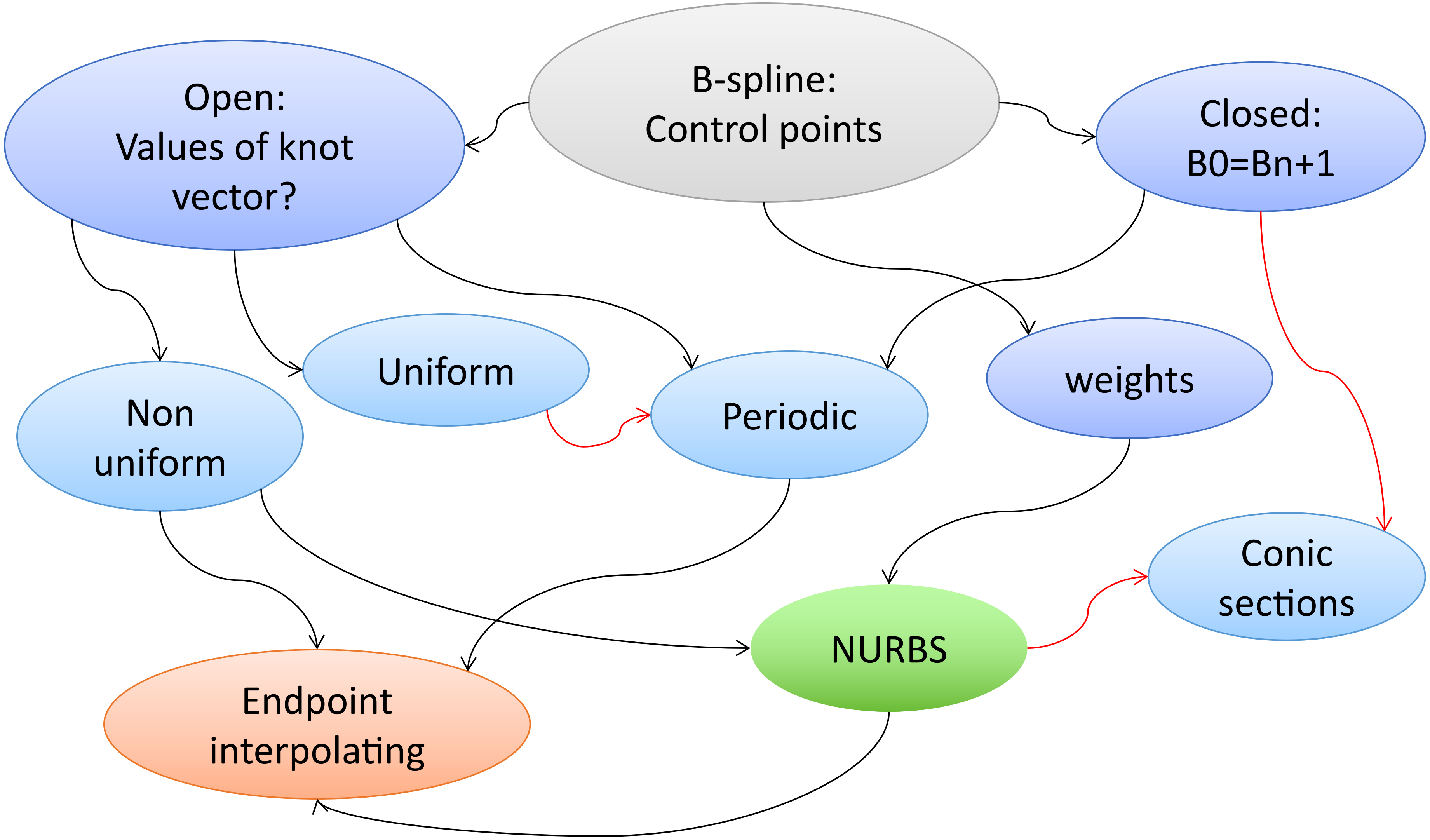




Three cubic B-spline curves.

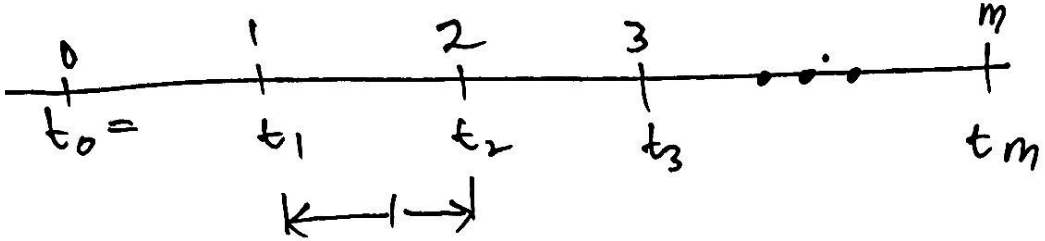


Types of B-Spline: based on knot vectors



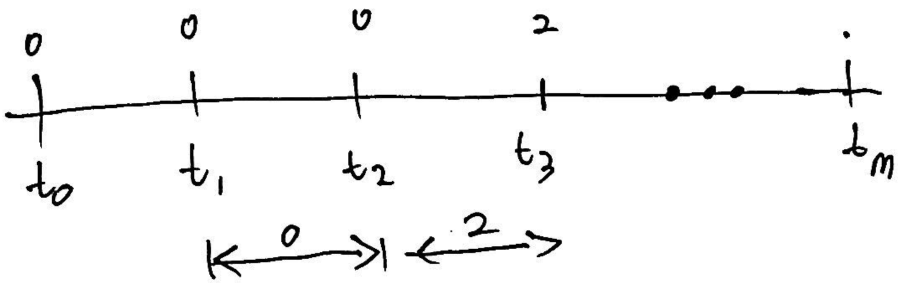
Types of knot vectors

Uniform B spline



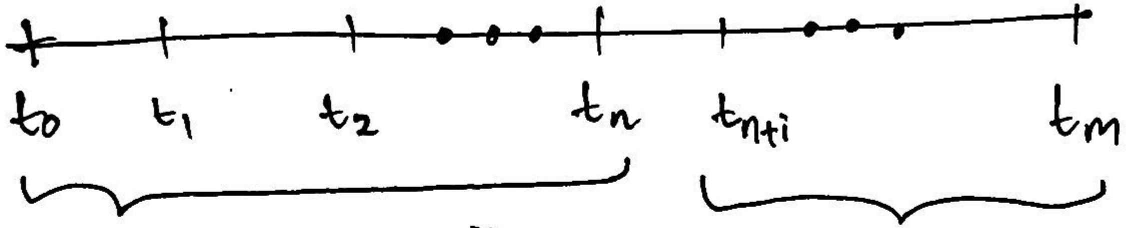
otherwise.

Nonuniform B spline



periodic B spline

d = degree

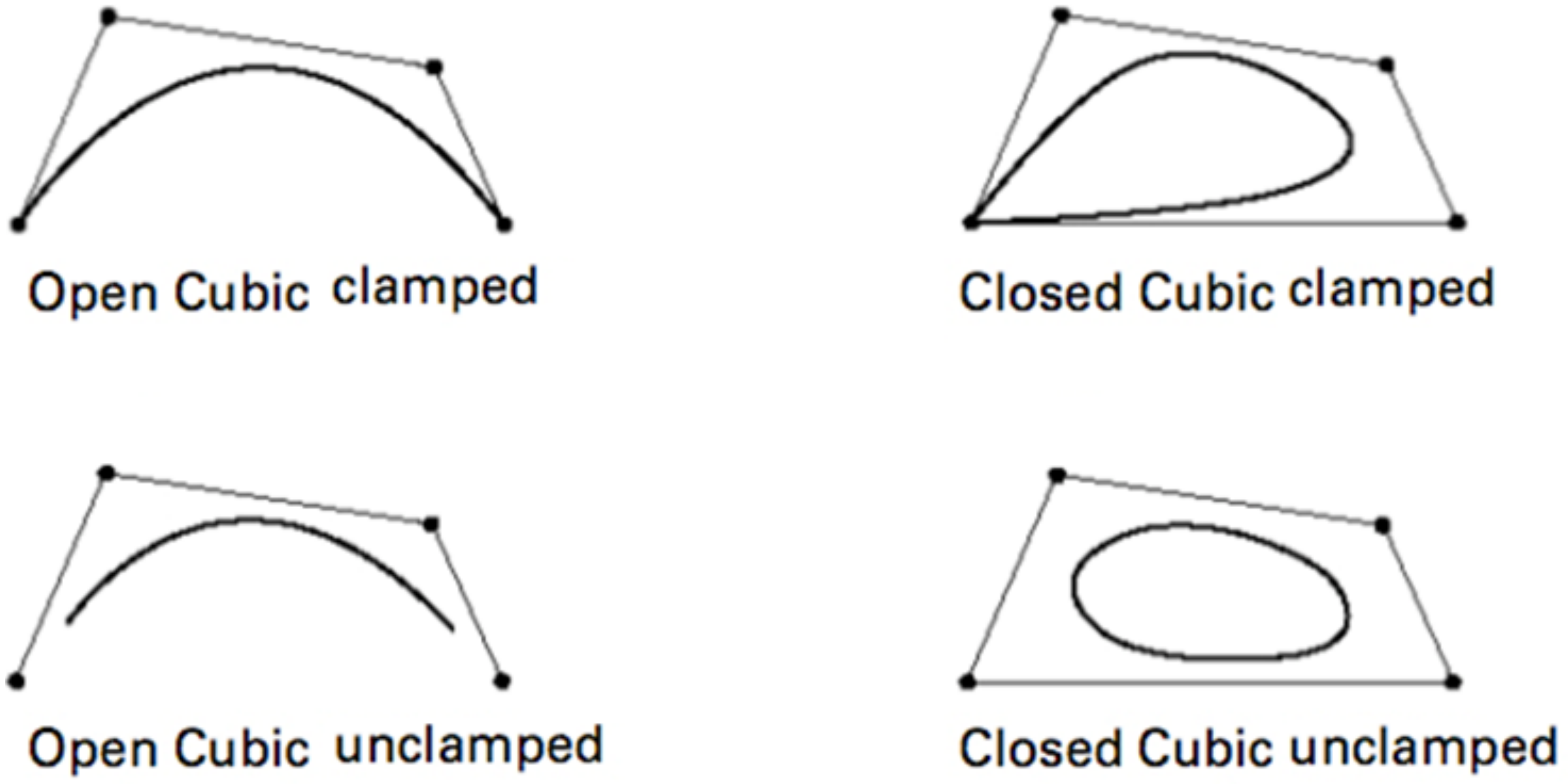


given as you like.

$$t_{n+i} = t_{n+i-1} + (t_i - t_{i-1})$$

$$i = 1, \dots, d+1.$$

Open & closed B-Spline



Definition 8.20

The NURBS curve of degree d (order $d+1$) with control points $\mathbf{b}_0, \ldots, \mathbf{b}_n$, weights $w_0, \ldots, w_n$, and knot vector $t_0, \ldots, t_m$, is the curve defined on the interval $[a, b] = [t_d, t_{m-d}]$ given by

$$\mathbf{B}(t) = \frac{\sum_{i=0}^{n} w_i \mathbf{b}_i N_{i,d}(t)}{\sum_{i=0}^{n} w_i N_{i,d}(t)} , \tag{8.13}$$

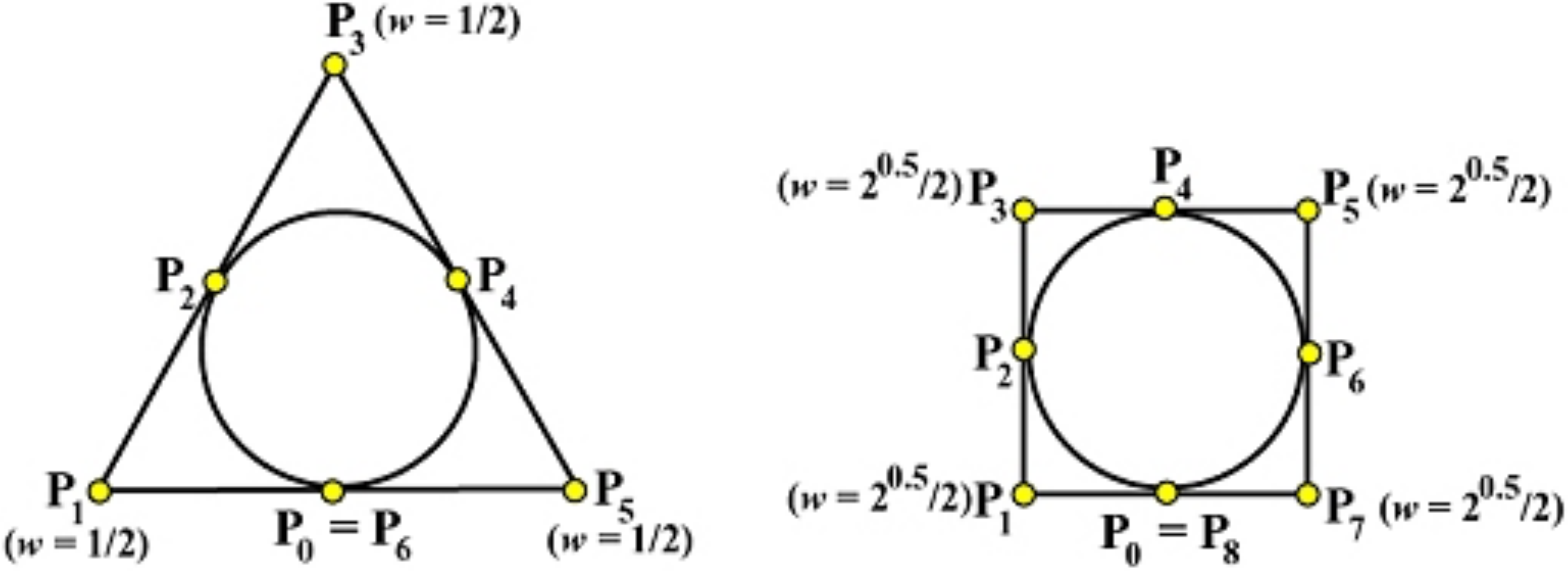
where $N_{i,d}(t)$ are the B-spline basis functions defined on the specified knot vector, and with the understanding that if $w_i = 0$ then $w_i \mathbf{b}_i$ is to be replaced by $\mathbf{b}_i$. The curve may also be written in the form

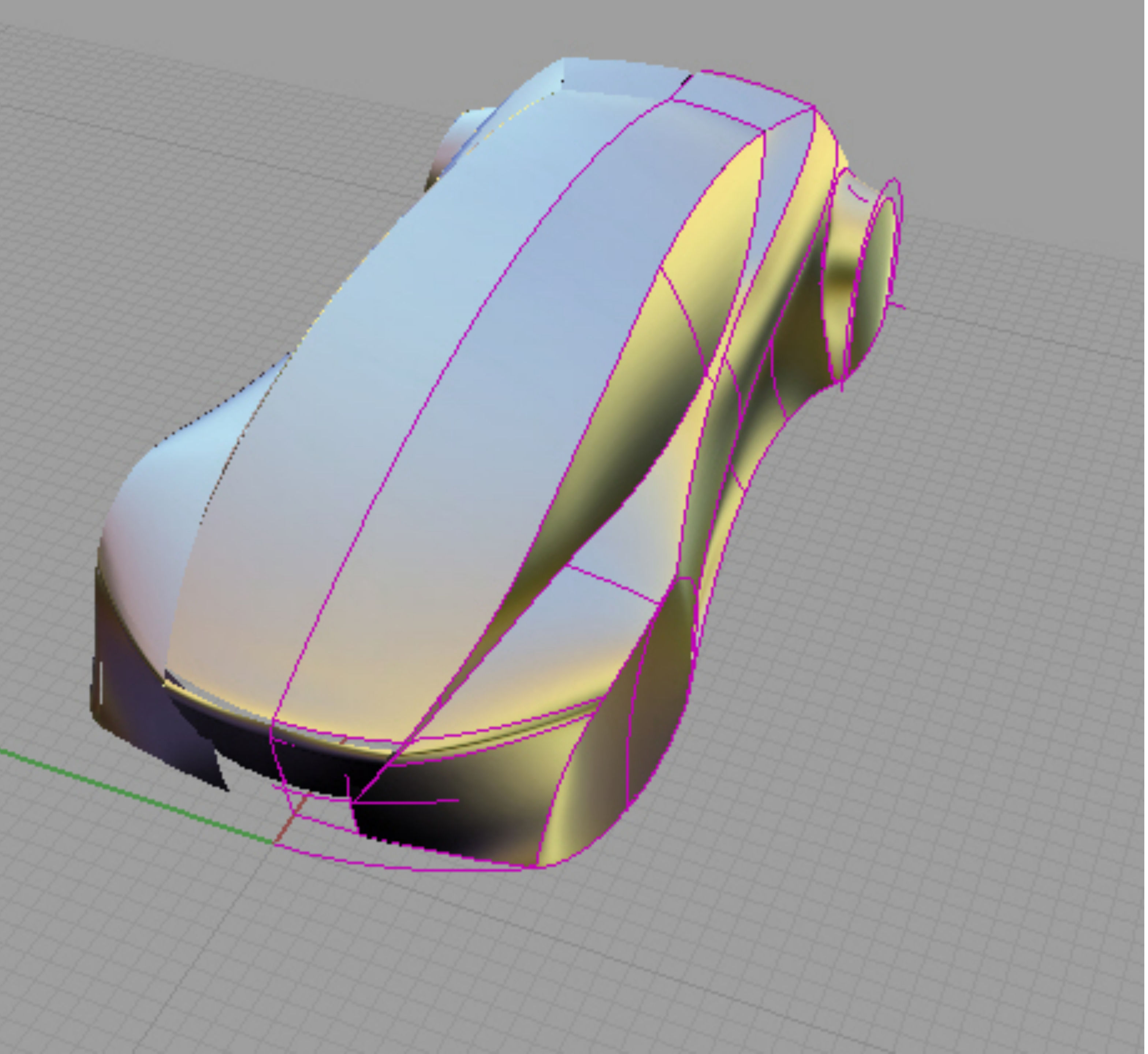
$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{b}_i R_{i,d}(t) ,$$

where

$$R_{i,d}(t) = \frac{w_i N_{i,d}(t)}{\sum_{j=0}^{n} w_j N_{j,d}(t)}$$

NURBS curves of degree 2 to represent a circle





SURFACES

Mathematics of shapes

B-Spline Surface

Let $N_{i,d}(s)$ be the B-spline basis functions of degree d with knot vector $s_0, s_1, \ldots, s_m$, and let $N_{j,e}(t)$ be the B-spline basis functions of degree e with knot vector $t_0, t_1, \ldots, t_q$. A *B-spline surface* with control points $\mathbf{p}_{i,j}$ $(0 \le i \le n = m - d - 1,\ 0 \le j \le p = q - e - 1)$ is defined by

$$\boxed{\mathbf{S}(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{p} \mathbf{p}_{i,j} N_{i,d}(s) N_{j,e}(t), \text{ for } (s,t) \in [s_d, s_{m-d}] \times [t_e, t_{q-e}]}. \tag{9.9}$$

A *NURBS surface* with control points $\mathbf{p}_{i,j}$ and weights $w_{i,j}$ is defined by

$$\mathbf{S}(s,t) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{p} w_{i,j} \mathbf{p}_{i,j} N_{i,d}(s) N_{j,e}(t)}{\sum_{i=0}^{n} \sum_{j=0}^{p} w_{i,j} N_{i,d}(s) N_{j,e}(t)}, \text{ for } (s,t) \in [s_d, s_{m-d}] \times [t_e, t_{q-e}]\,. \tag{9.10}$$

As for Bézier surfaces, the $(n+1) \times (p+1)$ control points of a B-spline or NURBS surface form a *control point polyhedron*. A B-spline surface is said to be *open* (respectively, *periodic*, *closed*) if the basis functions in both s and t are defined on *open* (respectively, *periodic*, *closed*) knot vectors.

Bézier or B-spline surfaces are said to be *bilinear*, *biquadratic*, *bicubic*, etc., whenever $n = p = 1$, $n = p = 2$, $n = p = 3$, etc.

Properties of B-Spline Surfaces

Theorem 9.11

A B-spline surface (9.9) satisfies the following properties.

Local Control: Each segment is determined by a $(d+1) \times (e+1)$ mesh of control points. If $s \in [s_\sigma, s_{\sigma+1})$ and $t \in [t_\tau, t_{\tau+1})$ $(d \le \sigma \le m-d-1,$ $e \le \tau \le n-e-1)$, then

$$\mathbf{S}(s,t) = \sum_{i=\sigma-d}^{\sigma} \sum_{j=\tau-e}^{\tau} \mathbf{p}_{i,j} N_{i,d}(s) N_{j,e}(t), \text{ for } (s,t) \in [s_d, s_{m-d}] \times [t_e, t_{q-e}] .$$

Convex Hull: If $s \in [s_\sigma, s_{\sigma+1})$ and $t \in [t_\tau, t_{\tau+1})$ $(d \le \sigma \le m-d-1,$ $e \le \tau \le n-e-1)$, then $\mathbf{S}(s,t) \in \mathrm{CH}\{\mathbf{p}_{\sigma-d,\tau-e}, \dots, \mathbf{p}_{\sigma,\tau}\}$.

Invariance under Affine Transformations: Let T be a three-dimensional affine transformation. Then

$$\mathsf{T}\left(\sum_{i=0}^{n} \sum_{j=0}^{p} \mathbf{p}_{i,j} N_{i,d}(s) N_{j,e}(t)\right) = \sum_{i=0}^{n} \sum_{j=0}^{p} \mathsf{T}\left(\mathbf{p}_{i,j}\right) N_{i,d}(s) N_{j,e}(t) .$$

Properties of NURBS Surfaces

Theorem 9.12

A NURBS surface (9.10) satisfies the following properties.

Local Control: If $s \in [s_\sigma, s_{\sigma+1})$ and $t \in [t_\tau, t_{\tau+1})$ $(d \leq \sigma \leq m-d-1,$ $e \leq \tau \leq n-e-1)$, then

$$\mathbf{S}(s,t) = \frac{\sum_{i=\sigma-d}^{\sigma} \sum_{j=\tau-e}^{\tau} w_{i,j} \mathbf{p}_{i,j} N_{i,d}(s) N_{j,e}(t)}{\sum_{i=\sigma-d}^{\sigma} \sum_{j=\tau-e}^{\tau} w_{i,j} N_{i,d}(s) N_{j,e}(t)} .$$

Convex Hull: If the weights w_i are all positive, then as for Theorem 9.11.

Invariance under Affine Transformations:

Let T be a three-dimensional affine transformation. Then

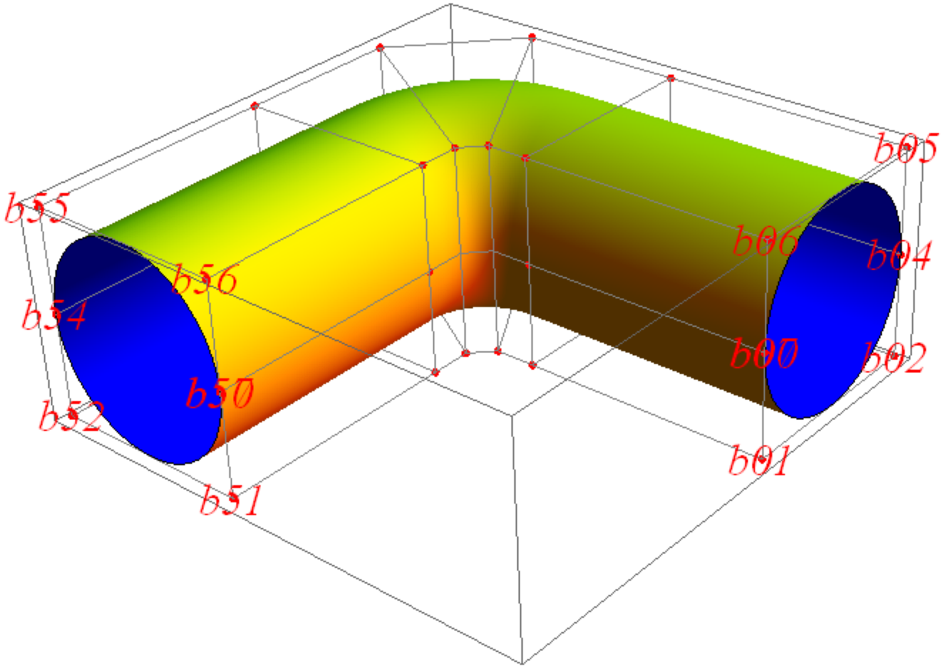
$$\begin{aligned} \mathsf{T}\left(\frac{\sum_{i=0}^{n} \sum_{j=0}^{p} w_{i,j} \mathbf{p}_{i,j} N_{i,d}(s) N_{j,e}(t)}{\sum_{i=0}^{n} \sum_{j=0}^{p} w_{i,j} N_{i,d}(s) N_{j,e}(t)} \right) \\ = \frac{\sum_{i=0}^{n} \sum_{j=0}^{p} w_{i,j} \mathsf{T}\left(\mathbf{p}_{i,j}\right) N_{i,d}(s) N_{j,e}(t)}{\sum_{i=0}^{n} \sum_{j=0}^{p} w_{i,j} N_{i,d}(s) N_{j,e}(t)} . \end{aligned}$$

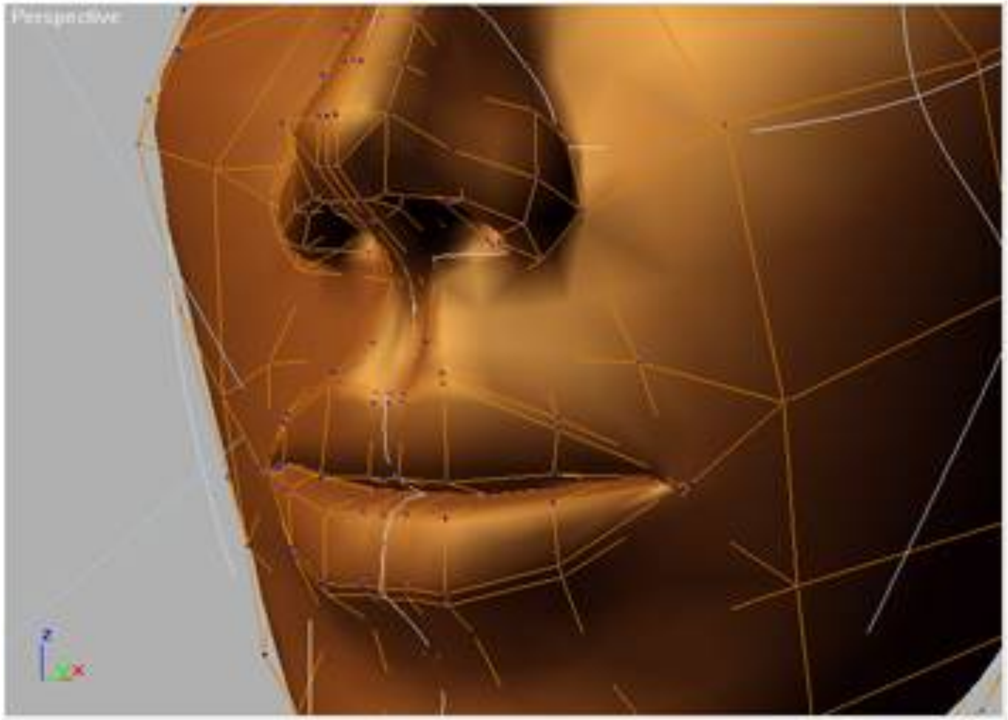
Invariance under Projective Transformations:

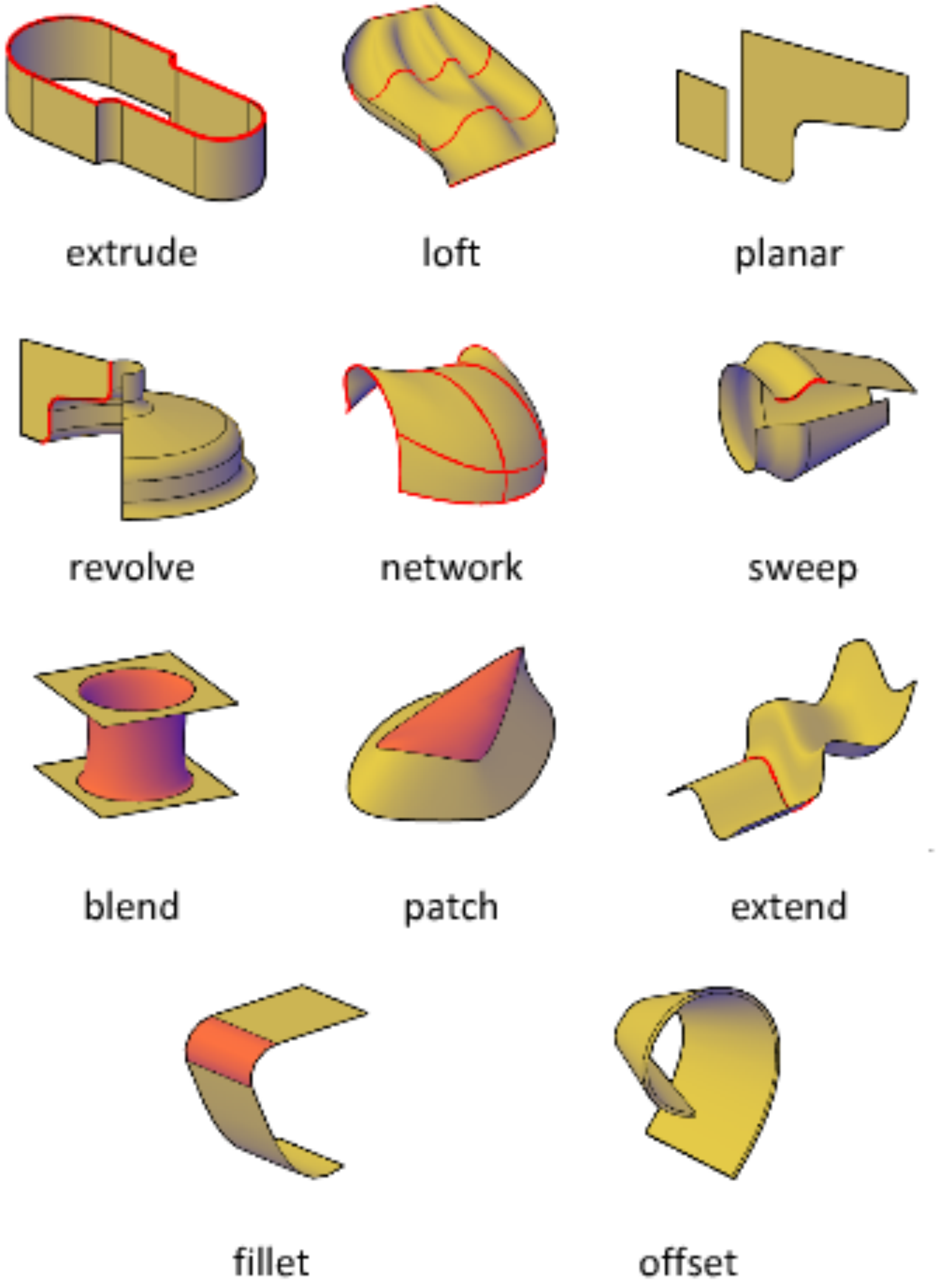
Let T be a three-dimensional projective transformation, and let $\hat{\mathbf{p}}_{i,j} = (w_{i,j}x_{i,j}, w_{i,j}y_{i,j}, w_{i,j}z_{i,j}, w_{i,j})$ be the homogeneous control points of $\mathbf{p}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$. Then

$$\mathsf{T}\left(\sum_{i=0}^{n} \sum_{j=0}^{p} \hat{\mathbf{p}}_{i,j} N_{i,d}(s) N_{j,e}(t) \right) = \sum_{i=0}^{n} \sum_{j=0}^{p} \mathsf{T}\left(\hat{\mathbf{p}}_{i,j}\right) N_{i,d}(s) N_{j,e}(t) .$$

Tensor Product: NURBS



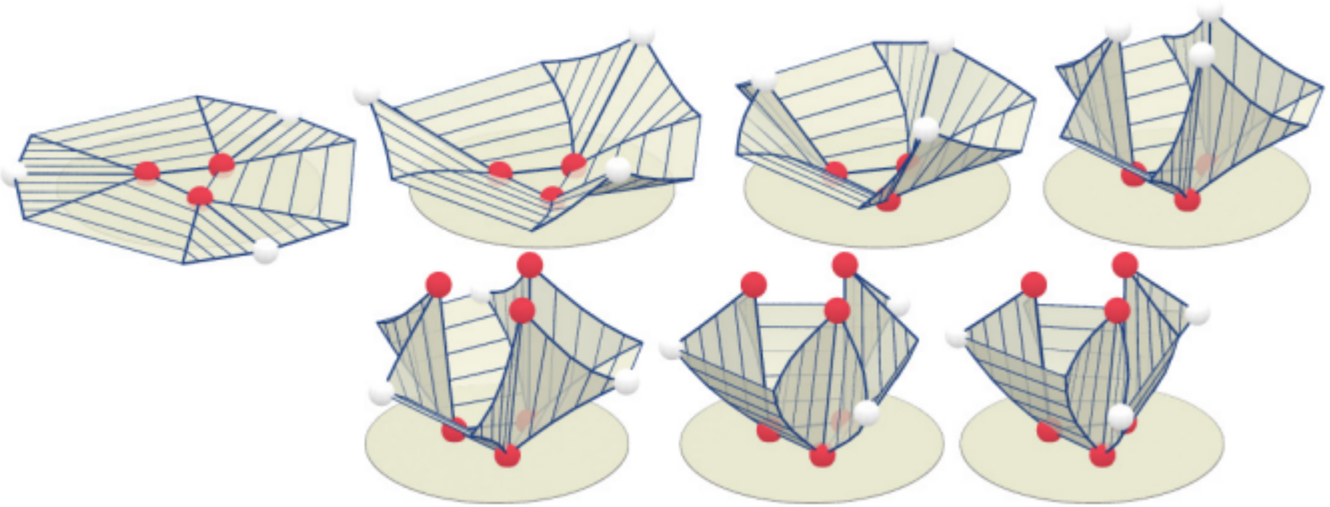


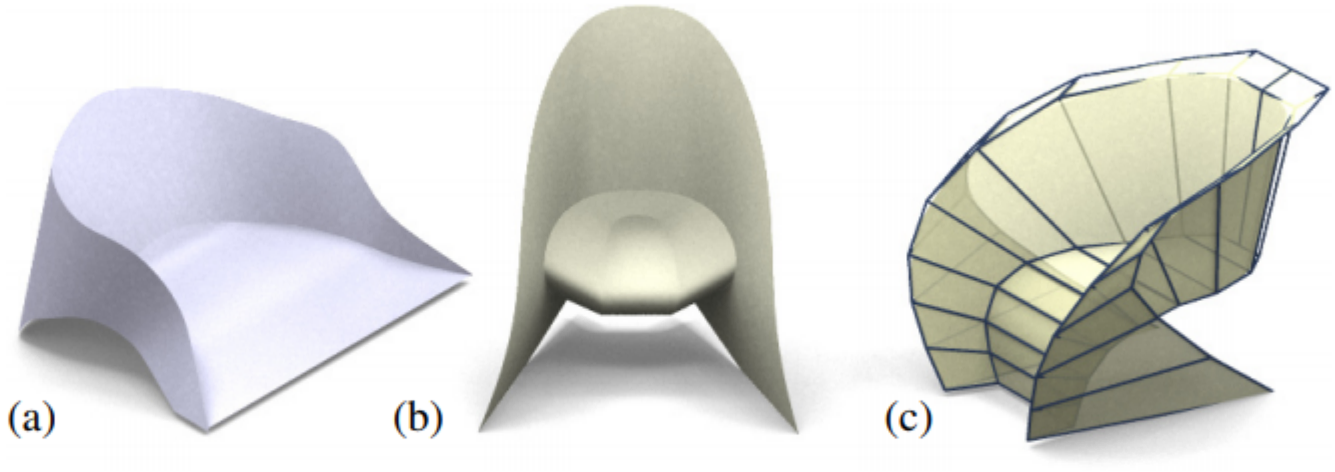


SURFACES

Mathematics of shapes

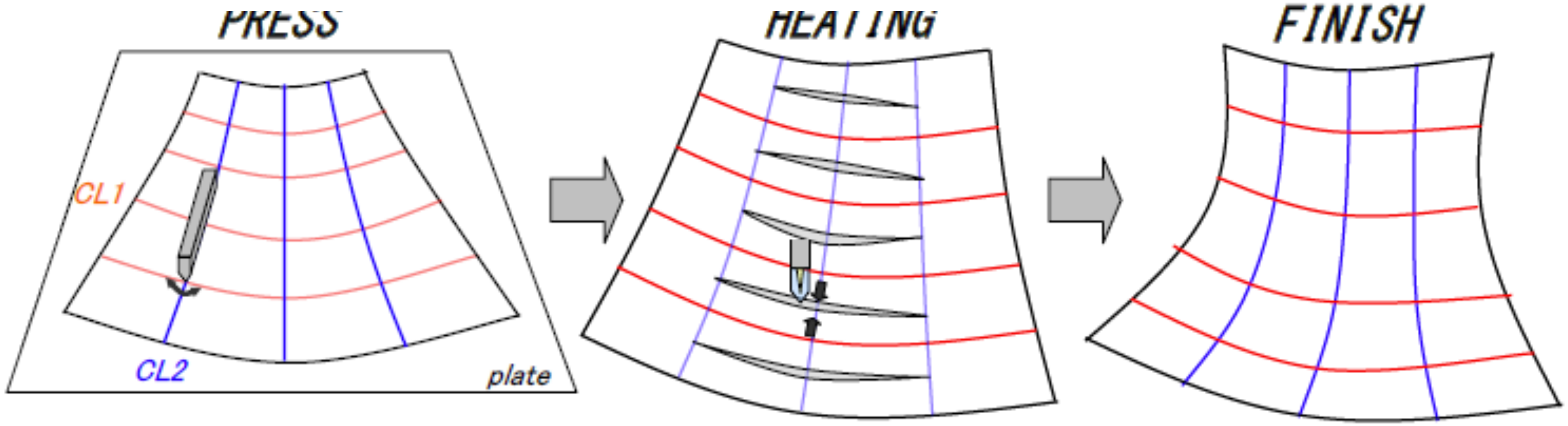
Developable surface





Undevelopable surfaces

Fig. 1: **Typical press work in Japanese shipyards**

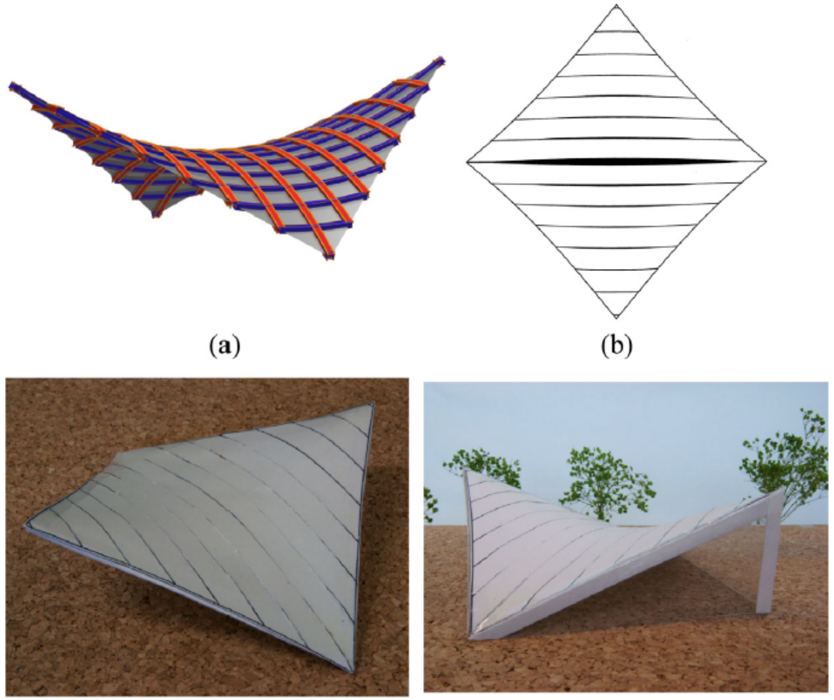


Press along the CL2
(i.e. bend strong curve parallel to CL1)

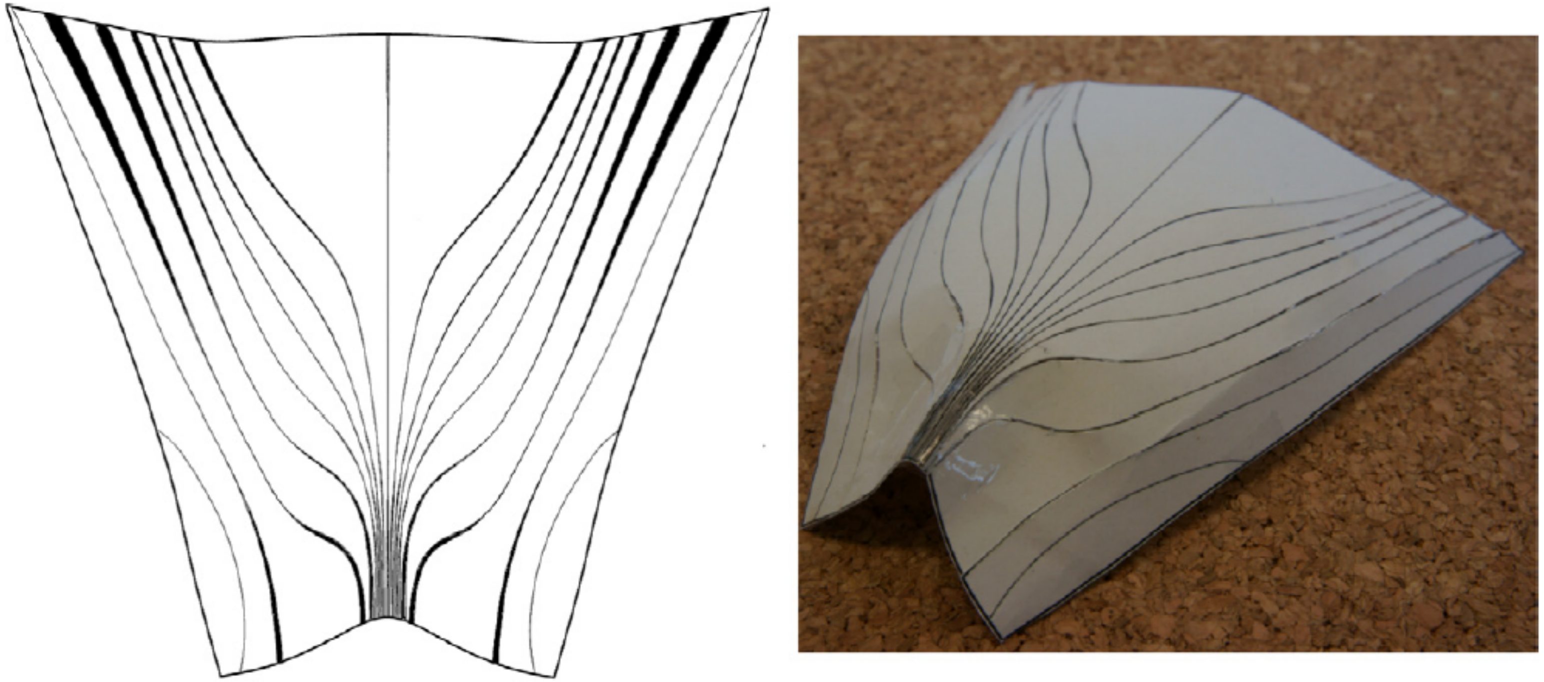
Heating along the CL1
(i.e. shrink parallel to CL2)
(i.e. forming an undevelopable surface)

Process of sheet metal forming using lines of curvature

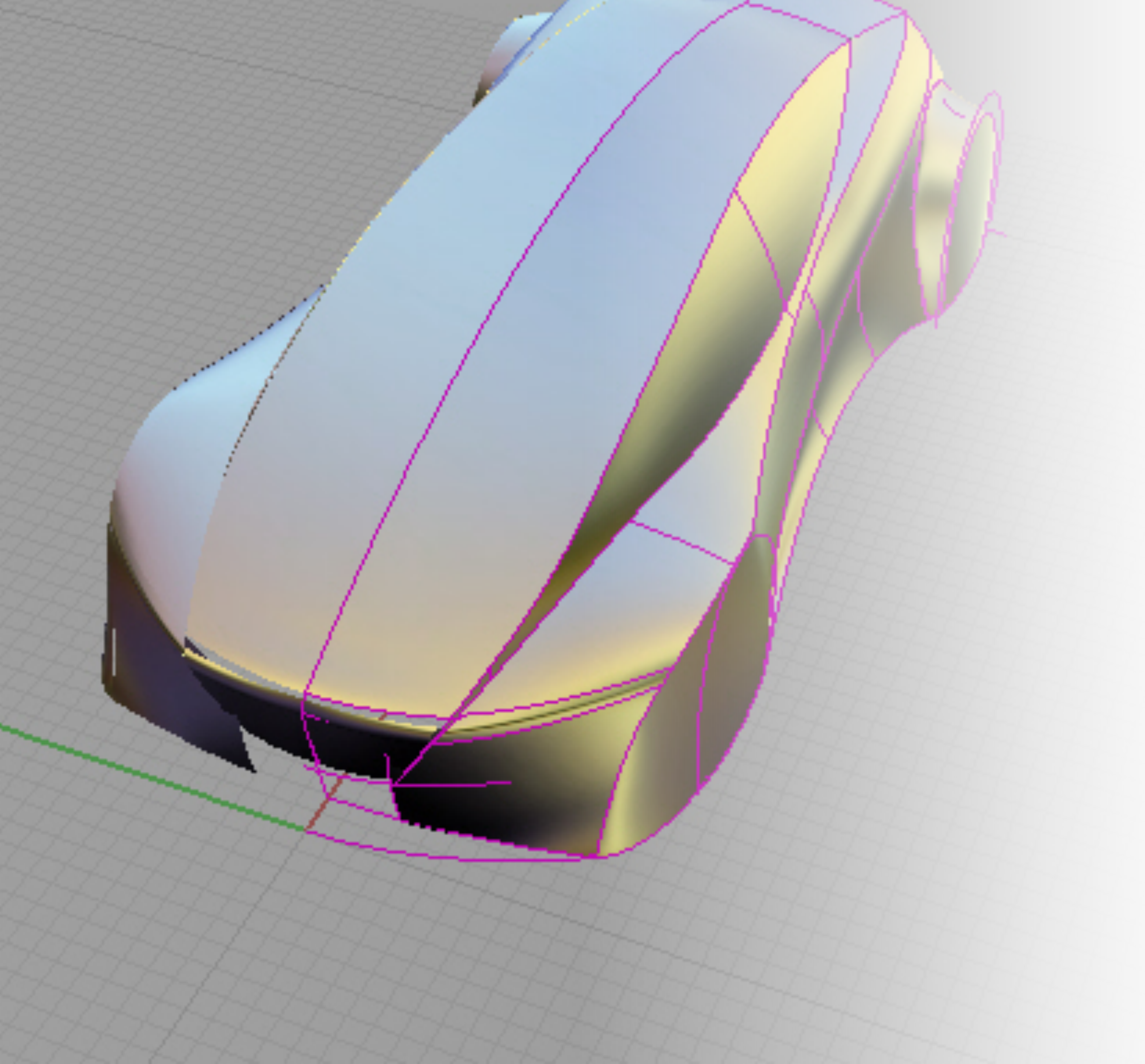
Undevelopable surfaces



(a) (b) (c) (d)



(a) (b)

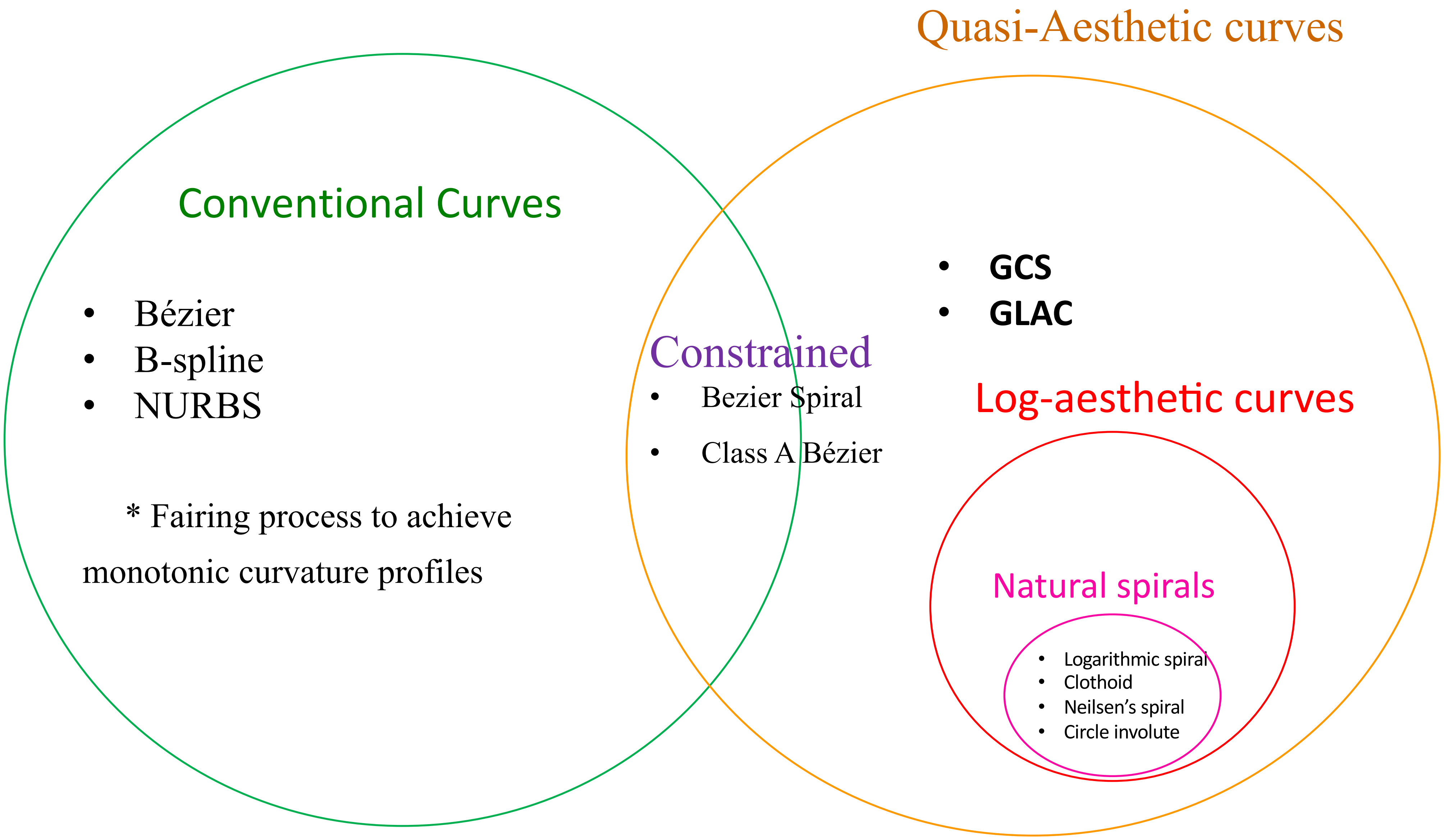


LACS

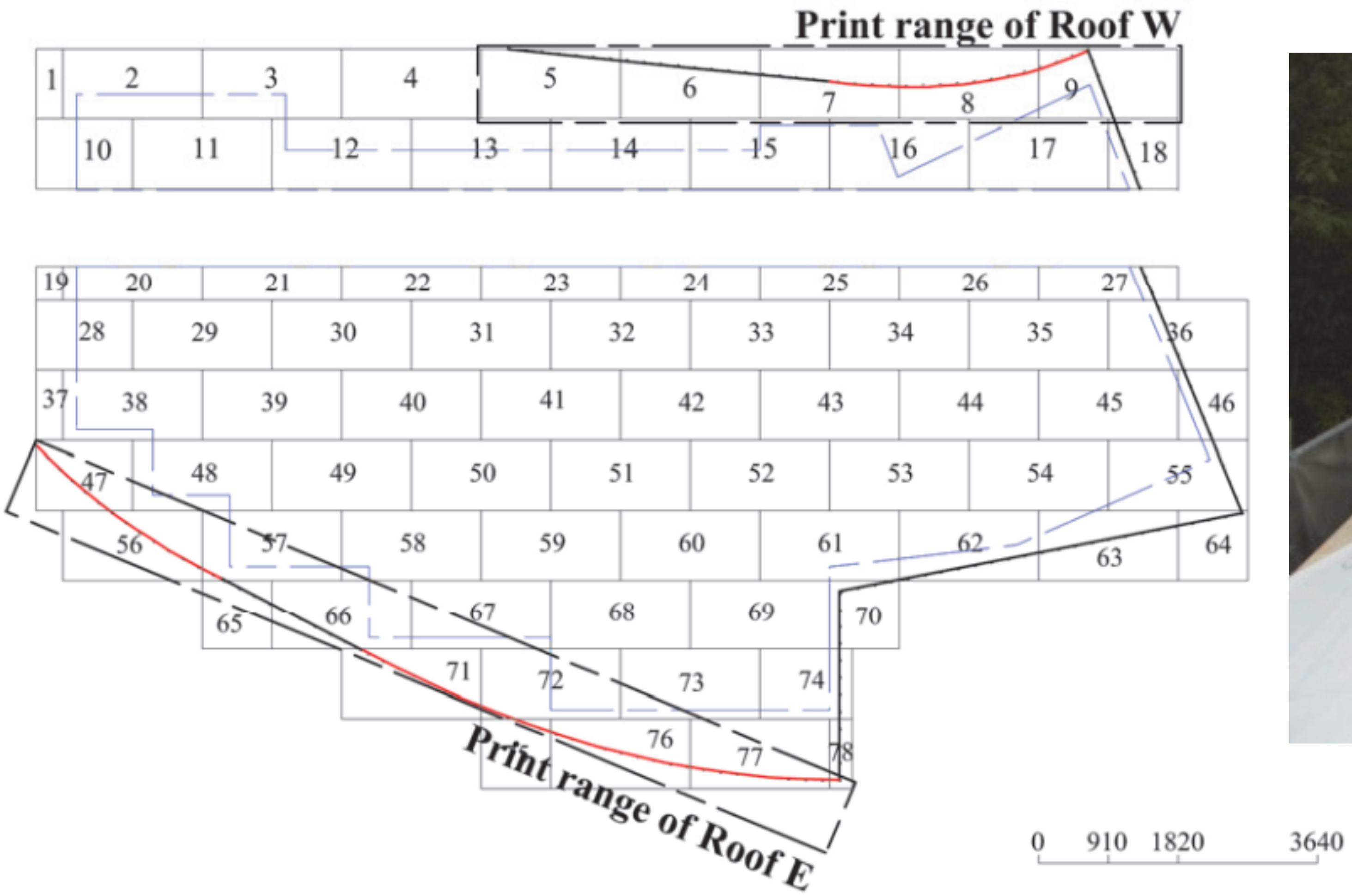
Towards aesthetic design

LAC developed similar to design practice!

Log-Aesthetic Curves (LACs)



House Design by T.Suzuki



APPLICATION OF LOG-AESTHETIC CURVES TO THE EAVES OF A WOODEN HOUSE www.mukogawa-u.ac.jp/~iasu2016/pdf/iaSU2016_proceedings_117.pdf



Path Planning: idea from highway design

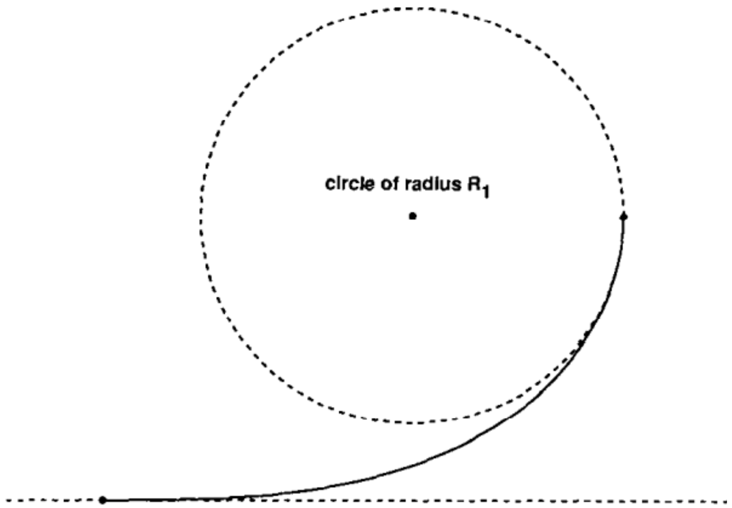


Fig. 1. Line to circle with a single spiral.

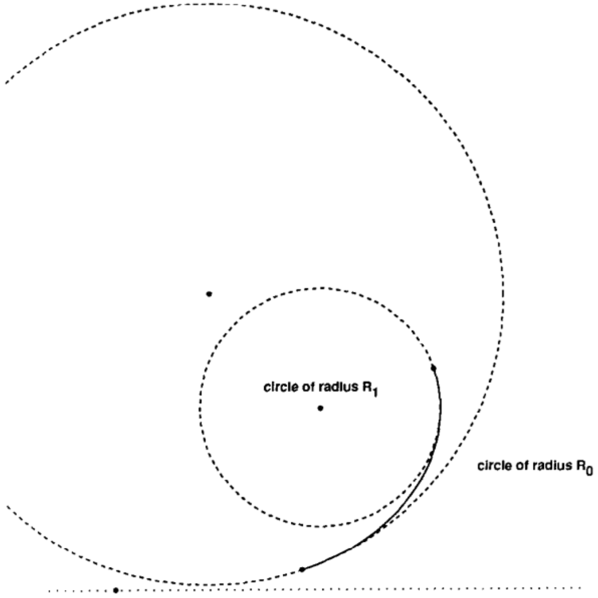
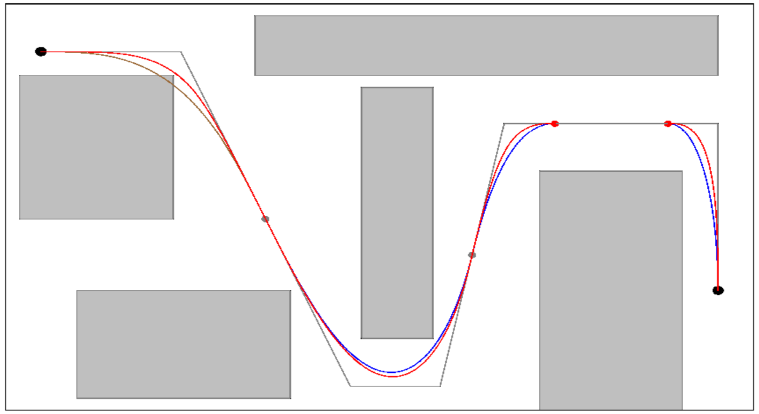


Fig. 2. Circle to circle with a single spiral.



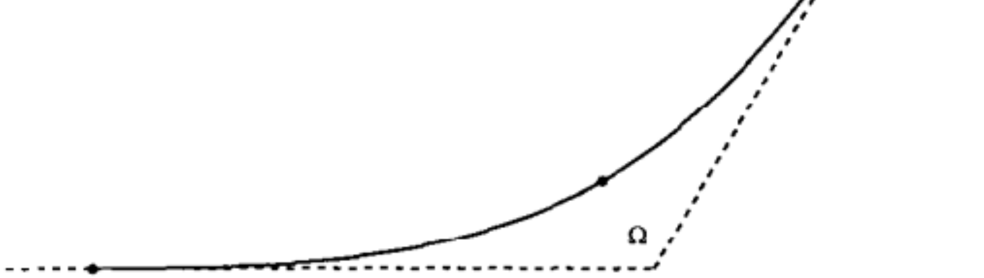


Fig. 5. Line to line with a pair of spirals.

bi-clothoid: an unique solution always exist

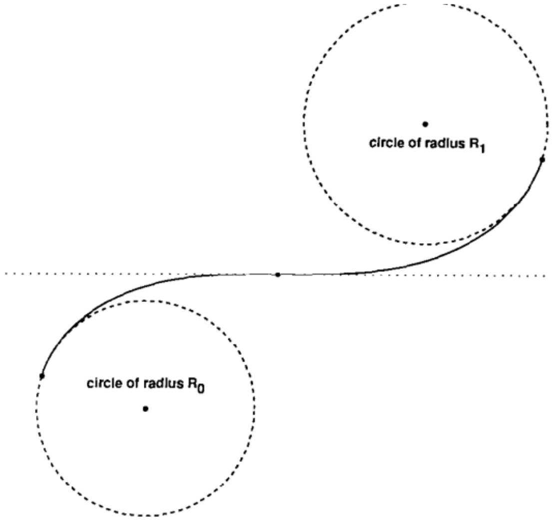


Fig. 3. Circle to circle forming an S-curve.

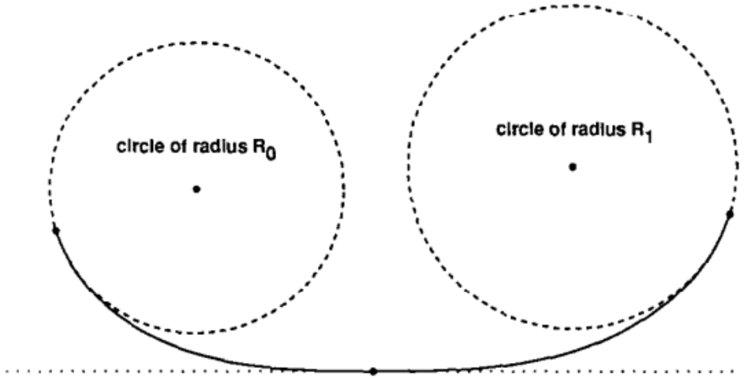


Fig. 4. Circle to circle forming a C-curve.

BAASS K. (1984),Transportation Forum, 1(3), 47–52

D.S. MEEK and D.J. WALTON (1989),Journal of Computational and Applied Mathematics 25 (1989) 69-78

Minimum variation log-aesthetic surfaces: approx. isoparametric curves into log-aesthetic curves.

$$J_{LAS} = \int_{u_1}^{u_2} \int_{v_1}^{v_2} \left(\frac{1}{\sqrt{E}} \alpha^2 (\rho^u)^{2\alpha-2} (\rho^u_u)^2 + \frac{1}{\sqrt{G}} \beta^2 (\rho^v)^{2\beta-2} (\rho^v_v)^2 \right) dv du \tag{6}$$

Here, $E$and G are elements of the first fundamental form and given by $E = \partial S/\partial u \cdot \partial S/\partial u$ and $G = \partial S/\partial v \cdot \partial S/\partial v$ respectively. ρ^u and ρ^v are radius of curvatures of isoparametric curve's u and v direction.

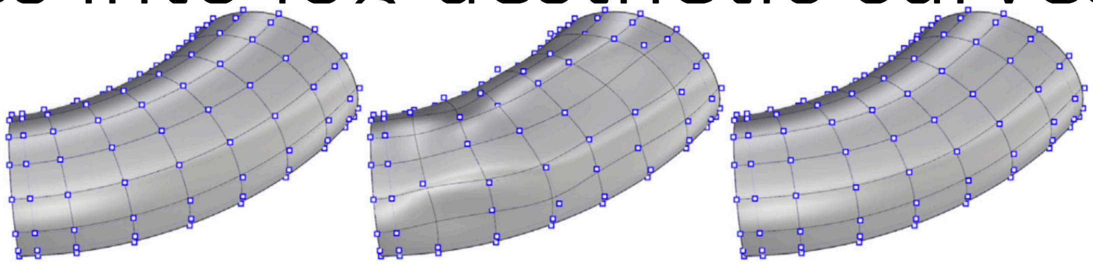


Fig. 6. Generated surfaces. Left: before optimization. Middle: surface with added noise. Right: after optimization.

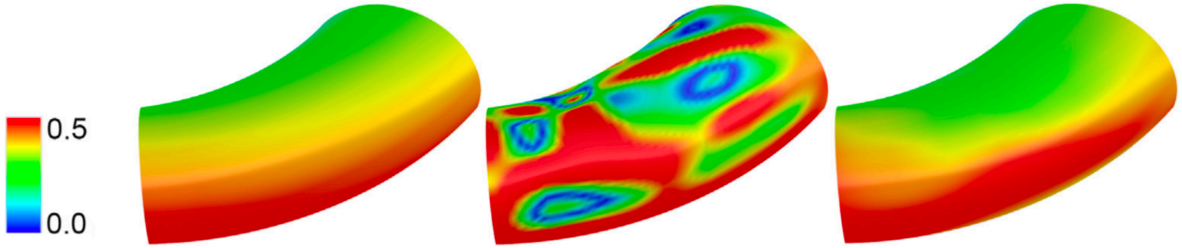


Fig. 7. Mean curvature distribution. Left: before optimization Middle: surface with added noise. Right: after optimization.

Fig. 8. Zebra map. Left: before optimization Middle: surface with added noise. Right: after optimization.

Sho Suzuki, R.U. Gobithaasan, Péter Salvi, Shin Usuki, Kenjiro T. Miura, Minimum variation log-aesthetic surfaces and their applications for smoothing free-form shapes, Journal of Computational Design and Engineering 5 (2018) 243–248

Thank You