Non-uniform Refine and Smooth Subdivision for General Degree B-splines

Tom Cashman[1] Neil Dodgson[1] Malcolm Sabin[2]

[1]Computer Laboratory
University of Cambridge

[2]Numerical Geometry Ltd.

8th November 2007

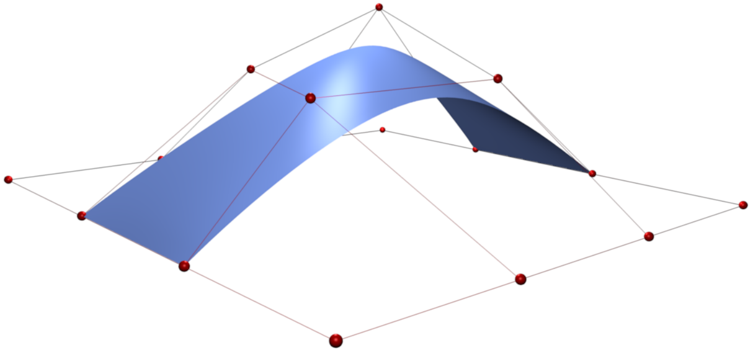
Outline

NURBS

- Non-Uniform Rational B-Splines
- NURBS curves are used in 2D (and 3D)

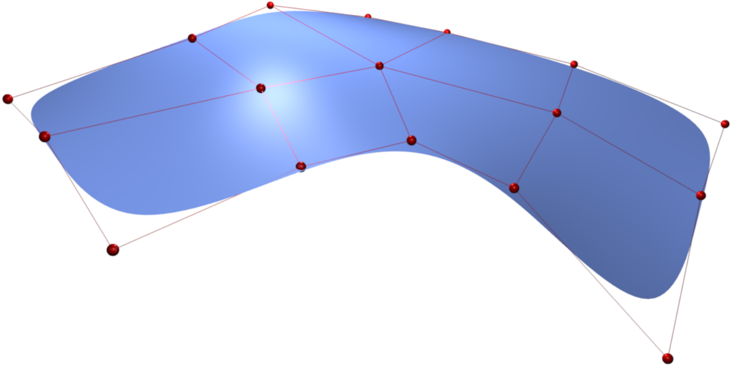
NURBS

- Non-Uniform Rational B-Splines
- NURBS curves are used in 2D (and 3D)
- NURBS surfaces use a rectangular control grid
- Industry standard for Computer-Aided Design (CAD)



Subdivision Surfaces

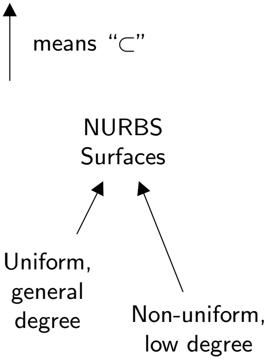
- Control meshes without a rigid rectangular grid
- Vertices with irregular valency are called extraordinary points
- Used heavily in animation since 'Geri's Game' (Pixar, 1997)



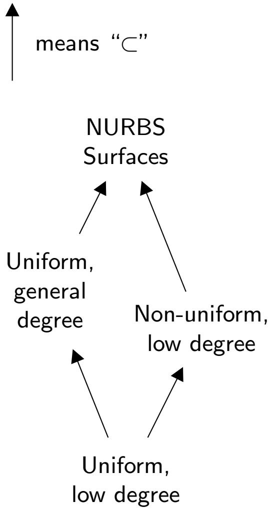
A Brief History of Subdivision Surfaces

NURBS
Surfaces

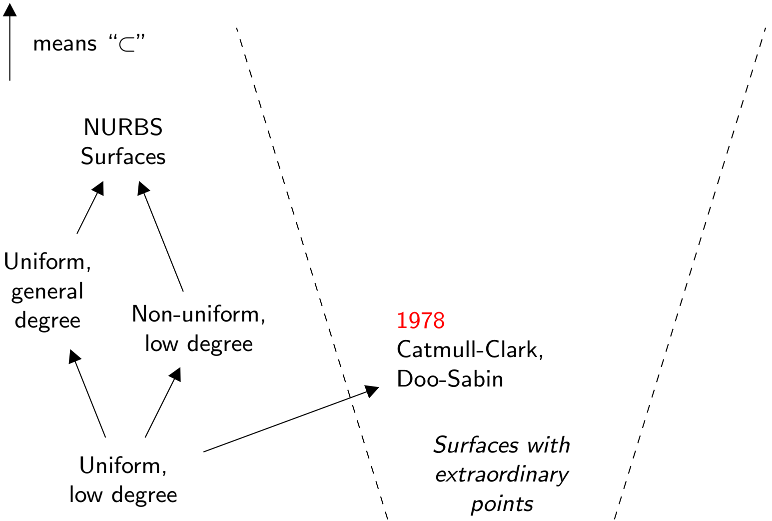
A Brief History of Subdivision Surfaces



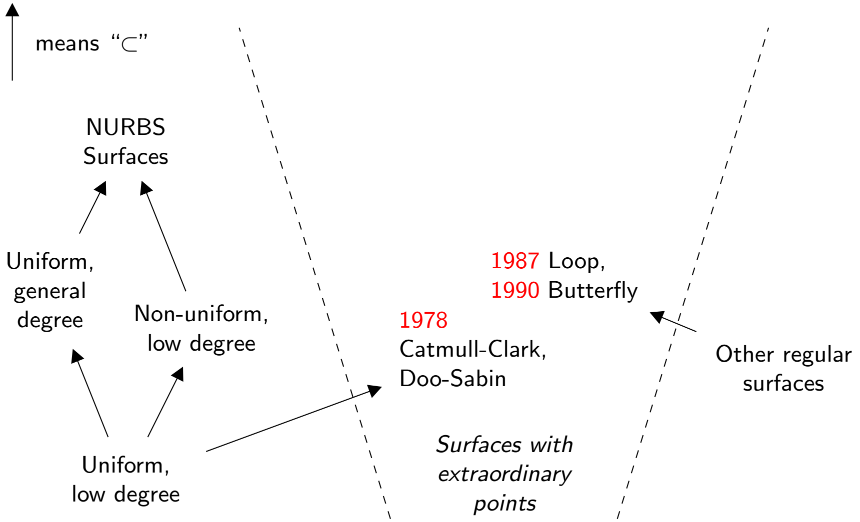
A Brief History of Subdivision Surfaces



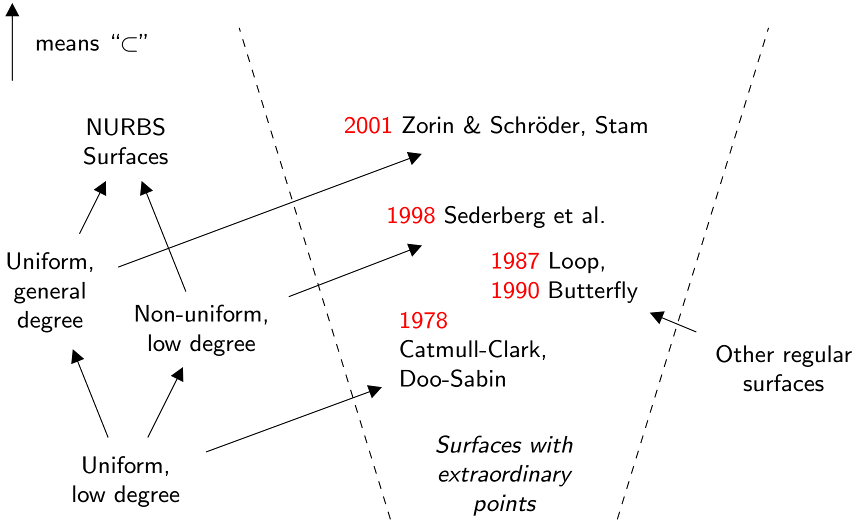
A Brief History of Subdivision Surfaces



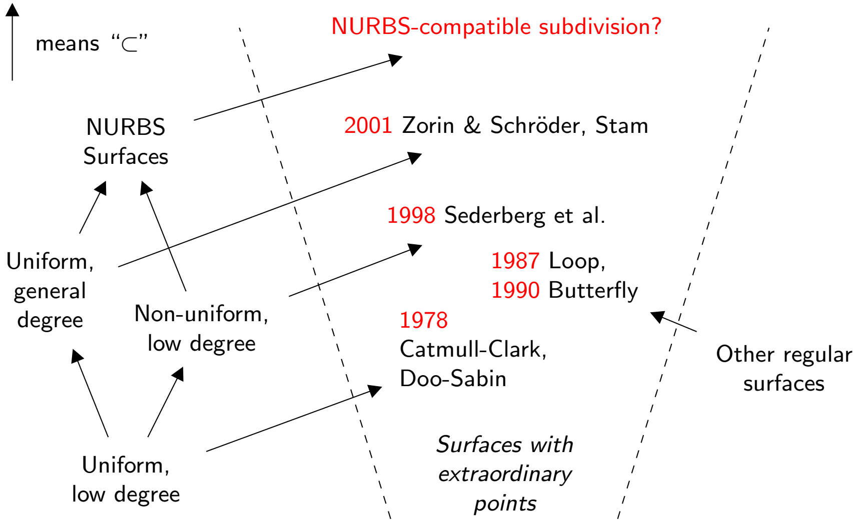
A Brief History of Subdivision Surfaces



A Brief History of Subdivision Surfaces

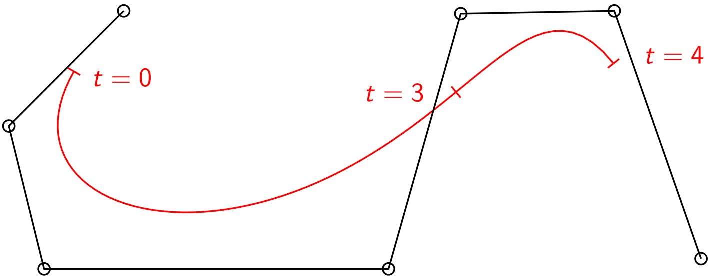


A Brief History of Subdivision Surfaces



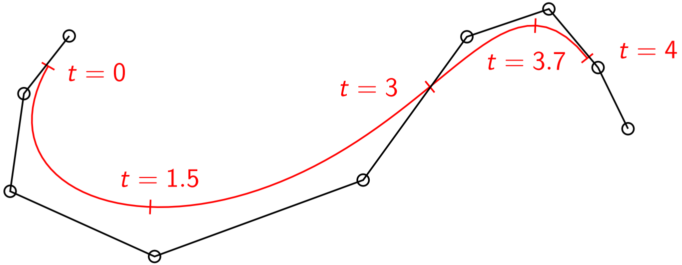
Knot Insertion

- B-splines are pieces of polynomial meeting at knots
- Uniform B-splines have even spacing between the knots
- Subdivision inserts more knots
  - Bringing the control polygon closer to the curve



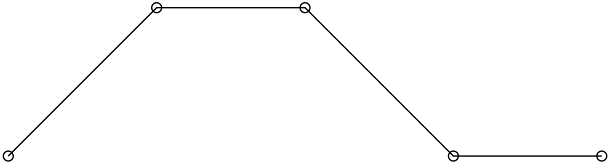
Knot Insertion

- B-splines are pieces of polynomial meeting at knots
- Uniform B-splines have even spacing between the knots
- Subdivision inserts more knots
  - Bringing the control polygon closer to the curve



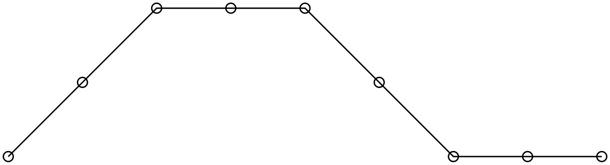
Refine and smooth subdivision

- Lane-Riesenfeld – uniform B-spline subdivision



Refine and smooth subdivision

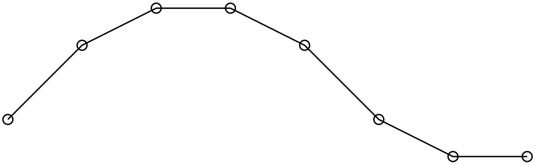
- Lane-Riesenfeld – uniform B-spline subdivision



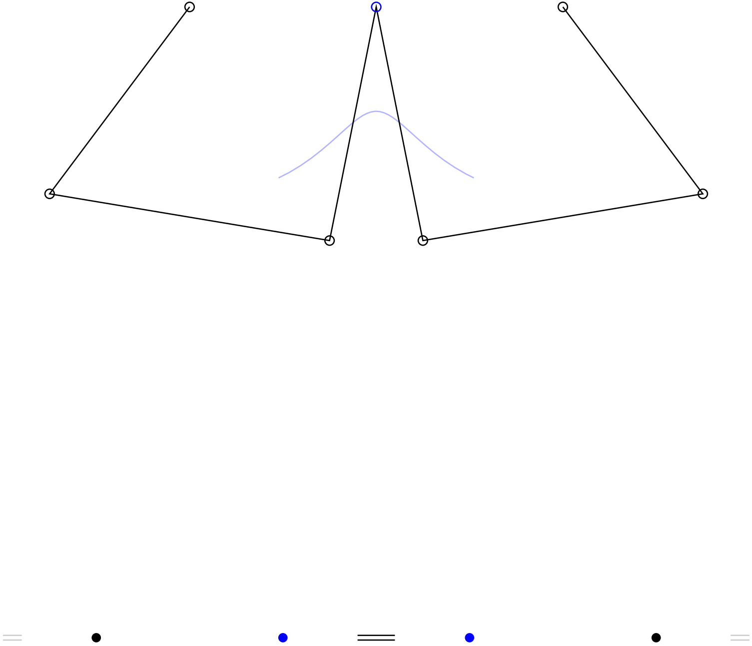
- Refine
  - polygon lengthened by adding points

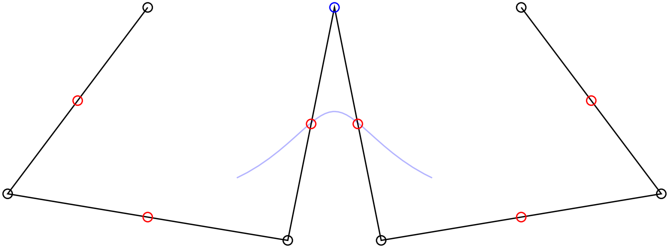
Refine and smooth subdivision

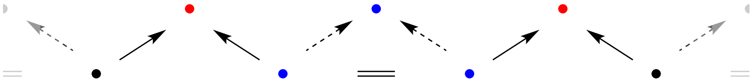
- Lane-Riesenfeld – uniform B-spline subdivision

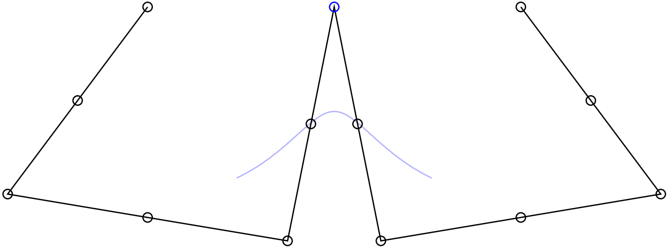


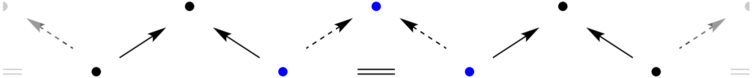
- Refine
  - polygon lengthened by adding points
- and Smooth
  - each step creates another polygon
  - points moved using local filters
- More smoothing steps for higher degree

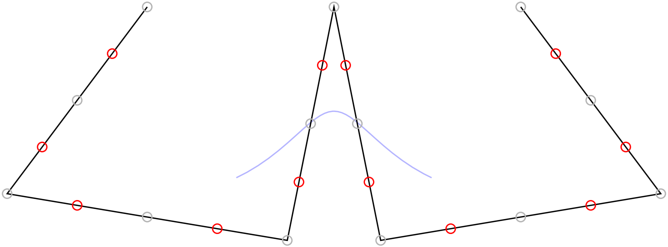


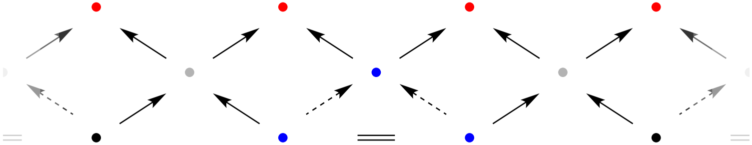


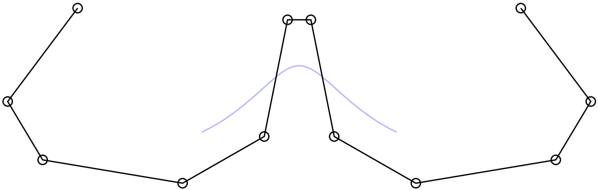


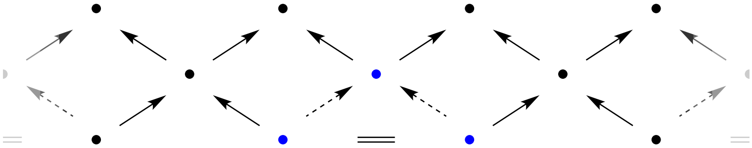


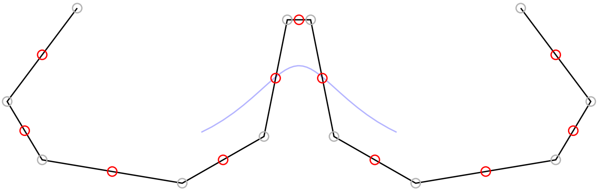


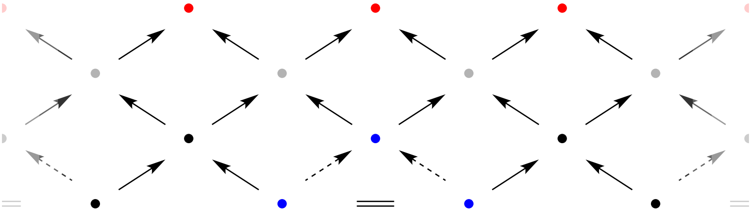


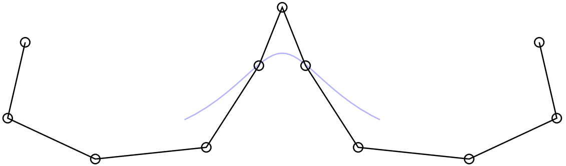


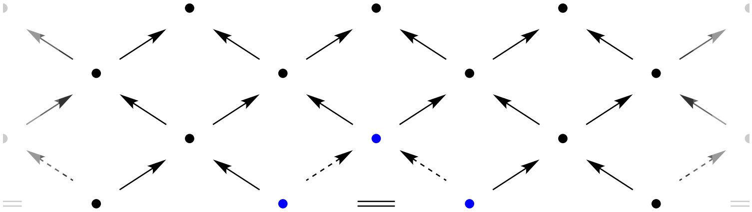


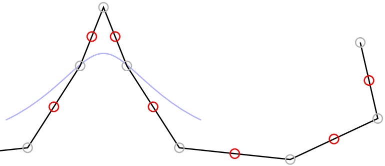


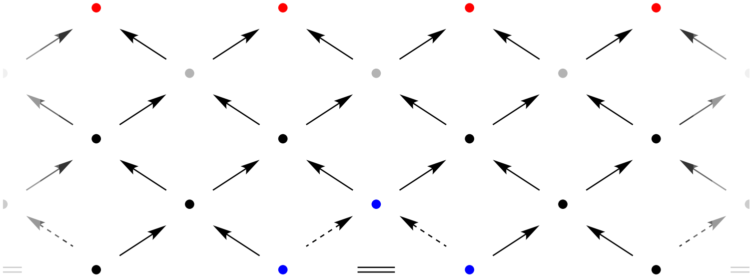


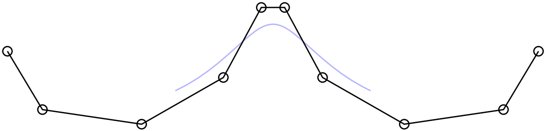


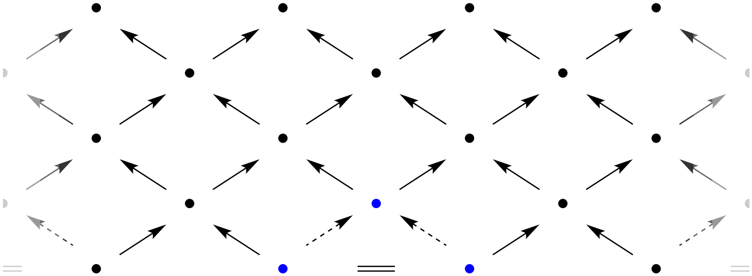


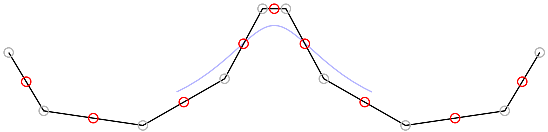


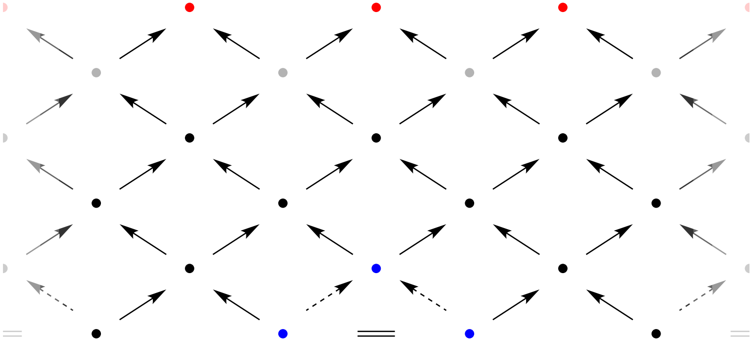


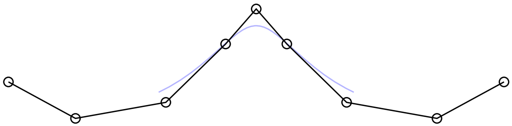


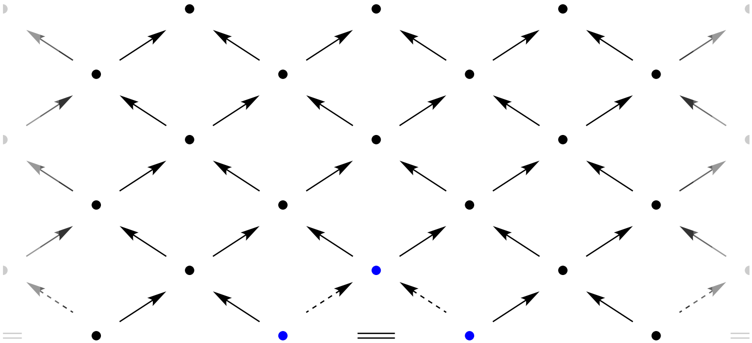




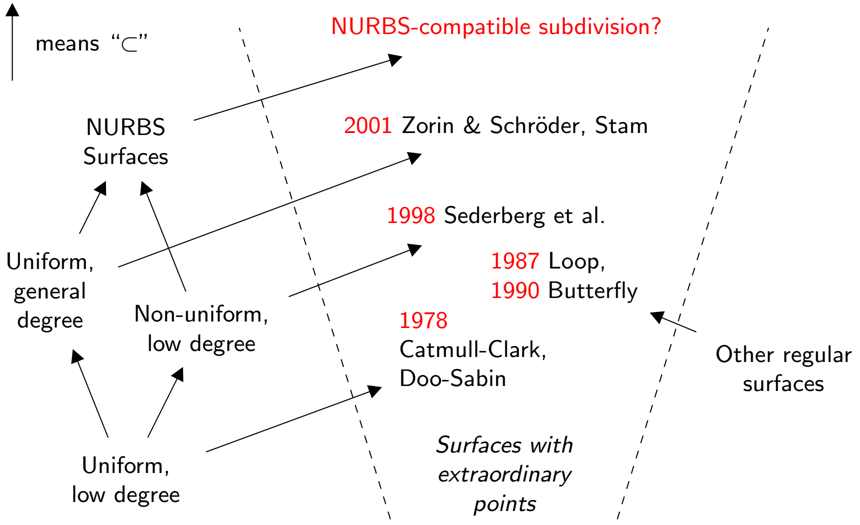








Why Refine and Smooth?



Problem statement

We want a knot insertion algorithm that is

- non-uniform,
- general degree, and uses
- refine and smooth

The polar form of a polynomial

Polynomials of degree d $\cong$ Symmetric d-affine maps *(polar form or blossom)*

The polar form of a polynomial

Polynomials of degree d	$\cong$	Symmetric d-affine maps *(polar form or blossom)*
$P(t)$	$=$	$p(t, t, \ldots, t, t)$

The polar form of a polynomial

Polynomials of degree d	$\cong$	Symmetric d-affine maps *(polar form or blossom)*
$P(t)$	$=$	$p(t, t, \ldots, t, t)$
$bx^2 + cx + d$	$=$	$b(x_1 x_2) + c(\frac{x_1+x_2}{2}) + d$

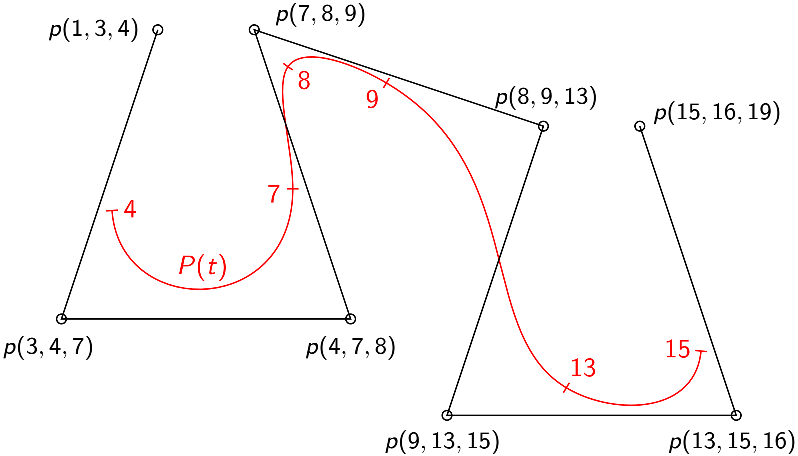
The polar form of a polynomial

Polynomials of degree d	$\cong$	Symmetric d-affine maps *(polar form or blossom)*
$P(t)$	$=$	$p(t, t, \ldots, t, t)$
$bx^2 + cx + d$	$=$	$b(x_1 x_2) + c(\frac{x_1+x_2}{2}) + d$
$ax^3 + bx^2 + cx + d$	$=$	$a(x_1 x_2 x_3) + b(\frac{x_1x_2+x_2x_3+x_1x_3}{3})$ $+c(\frac{x_1+x_2+x_3}{3}) + d$

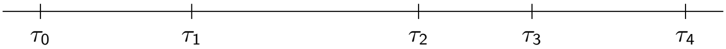
Properties: symmetric, multiaffine, diagonal

Blossoming and B-spline control points

Control points are the blossom evaluated at consecutive knots

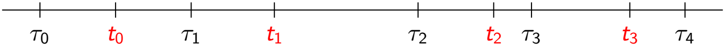


Blossoming and knot insertion



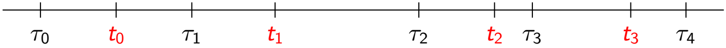
Blossoming and knot insertion

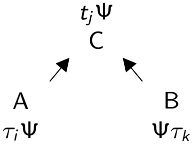
Insert knots t_i: $\tau_i \leq t_i \leq \tau_{i+1}$



Blossoming and knot insertion

Insert knots t_i: $\tau_i \leq t_i \leq \tau_{i+1}$

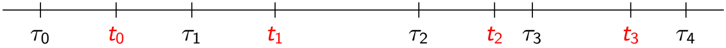


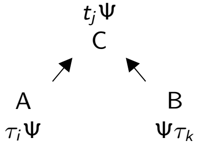


$$C = \frac{\tau_k - t_j}{\tau_k - \tau_i} A + \frac{t_j - \tau_i}{\tau_k - \tau_i} B$$

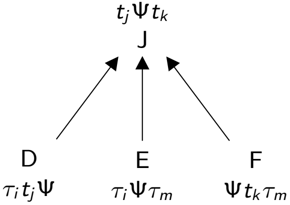
Blossoming and knot insertion

Insert knots t_i: $\tau_i \leq t_i \leq \tau_{i+1}$





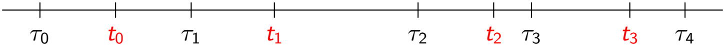
$$C = \frac{\tau_k - t_j}{\tau_k - \tau_i} A + \frac{t_j - \tau_i}{\tau_k - \tau_i} B$$

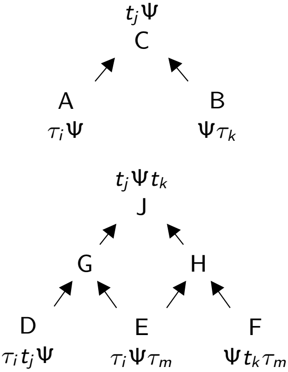


$$J = \frac{\tau_m - t_k}{\tau_m - \tau_i} D + \frac{t_k - t_j}{\tau_m - \tau_i} E + \frac{t_j - \tau_i}{\tau_m - \tau_i} F$$

Blossoming and knot insertion

Insert knots t_i: $\tau_i \leq t_i \leq \tau_{i+1}$





$$C = \frac{\tau_k - t_j}{\tau_k - \tau_i} A + \frac{t_j - \tau_i}{\tau_k - \tau_i} B$$

$$G = \frac{\tau_m - t_k}{\tau_m - x} D + \frac{t_k - x}{\tau_m - x} E$$

$$H = \frac{x - t_j}{x - \tau_i} E + \frac{t_j - \tau_i}{x - \tau_i} F$$

$$J = \frac{\tau_m - x}{\tau_m - \tau_i} G + \frac{x - \tau_i}{\tau_m - \tau_i} H$$

Outline

Adapting Lane-Riesenfeld

For Lane-Riesenfeld. . .

- Subdivision uses one refine and multiple smooth steps
- Smoothing filters compute a new point from two old points
- Smoothing filters compute the midpoint of two old points

Adapting Lane-Riesenfeld

For Lane-Riesenfeld...

- Subdivision uses one refine and multiple smooth steps
- Smoothing filters compute a new point from two old points
- Smoothing filters compute the midpoint of two old points
- Intermediate smoothing steps compute the subdivision result for lower degree

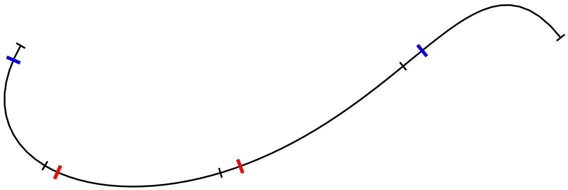
Adapting Lane-Riesenfeld

We want a non-uniform algorithm, where. . .

- Subdivision uses one refine and multiple smooth steps
- Smoothing filters compute a new point from two old points
- Smoothing filters compute the midpoint of two old points
- Intermediate smoothing steps compute the subdivision result for lower degree

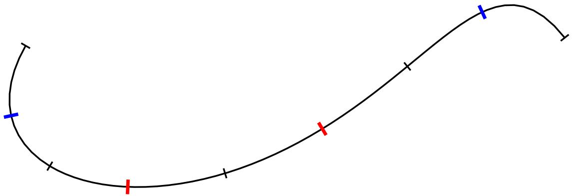
Maintaining degree independence

- But intermediate polygons that compute lower degree subdivision introduce constraints on $\boldsymbol{\tau}$ and $\mathbf{t}$



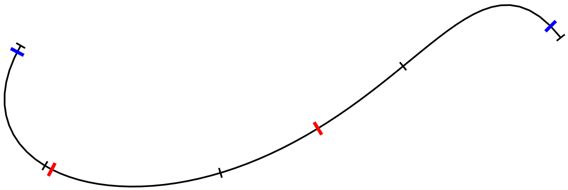
Maintaining degree independence

- But intermediate polygons that compute lower degree subdivision introduce constraints on $\boldsymbol{\tau}$ and $\mathbf{t}$



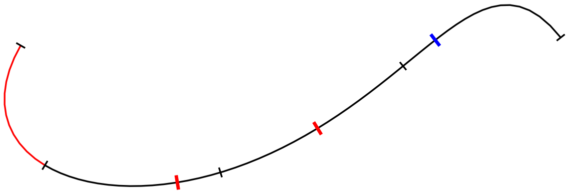
Maintaining degree independence

- But intermediate polygons that compute lower degree subdivision introduce constraints on $\boldsymbol{\tau}$ and $\mathbf{t}$



Maintaining degree independence

- But intermediate polygons that compute lower degree subdivision introduce constraints on $\boldsymbol{\tau}$ and $\mathbf{t}$



Revisiting our requirements

We want a non-uniform algorithm, where...

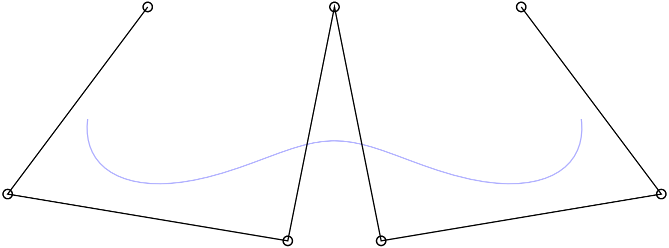
- Subdivision uses one refine and multiple smooth steps
- Smoothing filters compute a new point from two old points

- Intermediate smoothing steps compute the subdivision result for lower degree

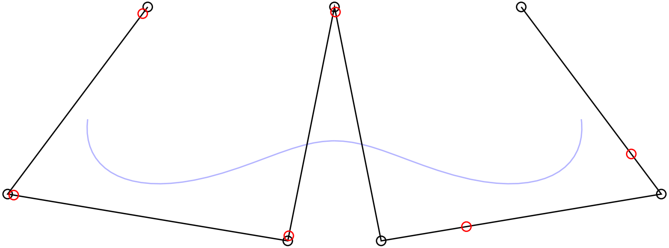
Revisiting our requirements

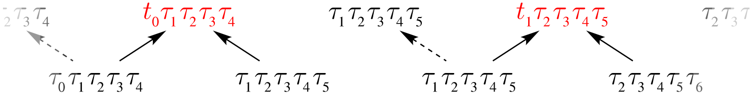
We want a non-uniform algorithm, where...

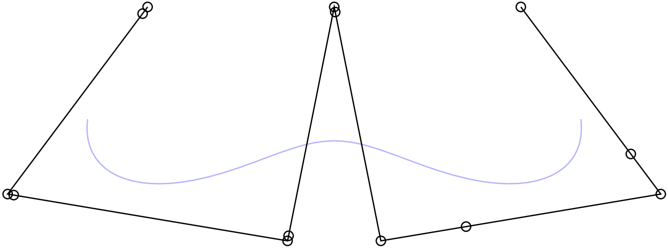
- Subdivision uses one refine and multiple smooth steps
- Smoothing filters compute a new point from two old points
- Intermediate smoothing steps compute the subdivision result for lower degree



$\mathcal{T}_0\,\mathcal{T}_1\,\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4$ $\quad\mathcal{T}_1\,\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4\,\mathcal{T}_5$ $\quad\mathcal{T}_1\,\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4\,\mathcal{T}_5$ $\quad\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4\,\mathcal{T}_5\,\mathcal{T}_6$

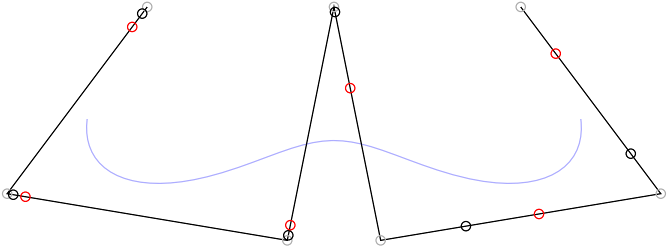


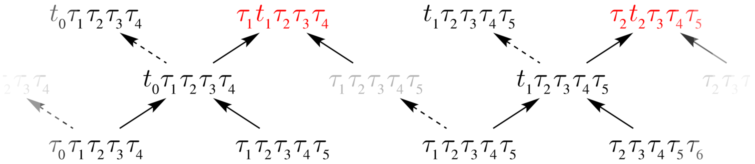

$\tau_2\tau_3\tau_4$
$t_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$t_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$



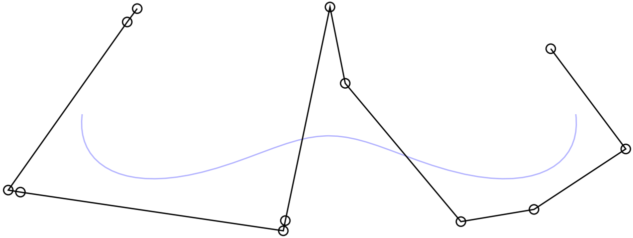
$_2\tau_3\tau_4$ $t_0\tau_1\tau_2\tau_3\tau_4$ $\tau_1\tau_2\tau_3\tau_4\tau_5$ $t_1\tau_2\tau_3\tau_4\tau_5$ $\tau_2\tau_3\tau$

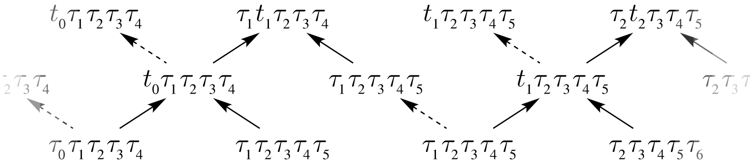
$\tau_0\tau_1\tau_2\tau_3\tau_4$ $\tau_1\tau_2\tau_3\tau_4\tau_5$ $\tau_1\tau_2\tau_3\tau_4\tau_5$ $\tau_2\tau_3\tau_4\tau_5\tau_6$

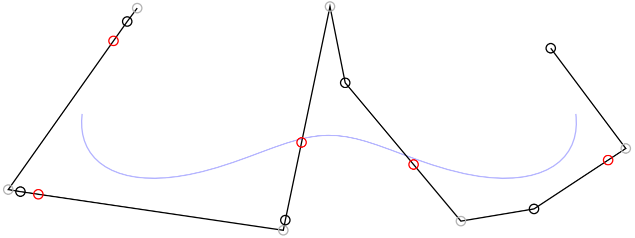


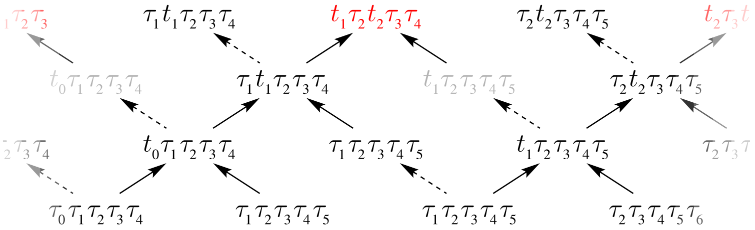


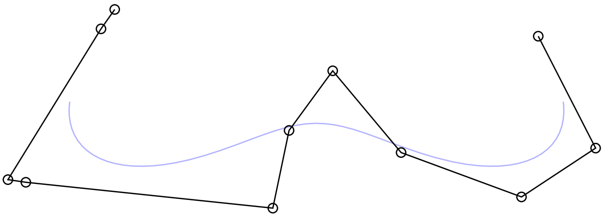
t0τ1τ2τ3τ4
τ1t1τ2τ3τ4
t1τ2τ3τ4τ5
τ2t2τ3τ4τ5
t0τ1τ2τ3τ4
τ1τ2τ3τ4τ5
t1τ2τ3τ4τ5
τ0τ1τ2τ3τ4
τ1τ2τ3τ4τ5
τ1τ2τ3τ4τ5
τ2τ3τ4τ5τ6

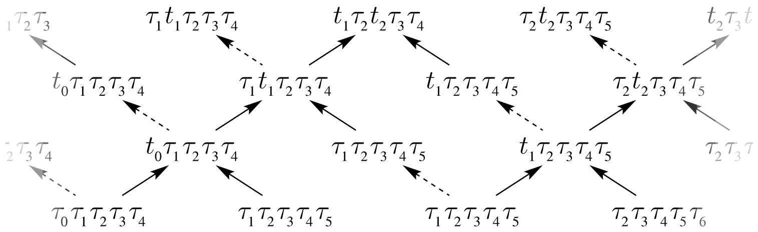


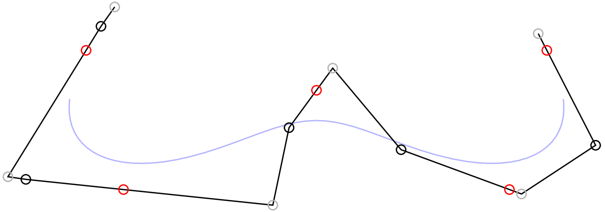

$t_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1 t_1\tau_2\tau_3\tau_4$
$t_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2 t_2\tau_3\tau_4\tau_5$
$t_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$t_1\tau_2\tau_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

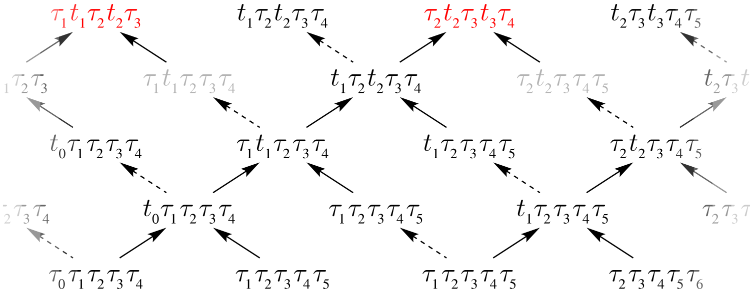


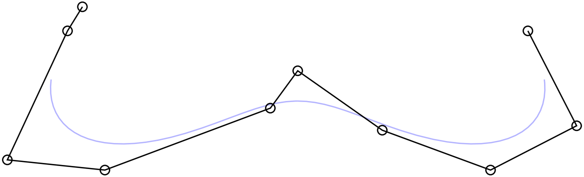

$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 t_2 \tau_3 \tau_4$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 \tau_3 \tau_4 \tau_5 \tau_6$

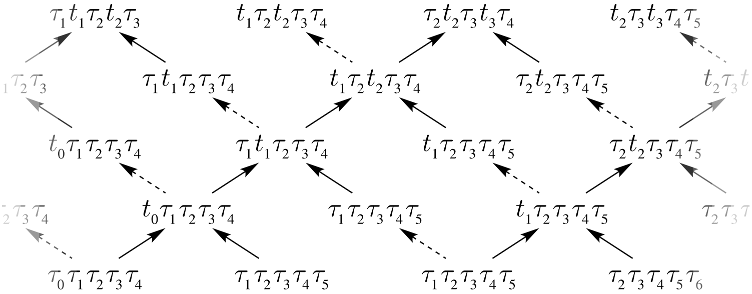


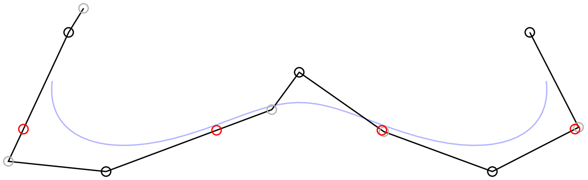

$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 t_2 \tau_3 \tau_4$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 \tau_3 \tau_4 \tau_5 \tau_6$

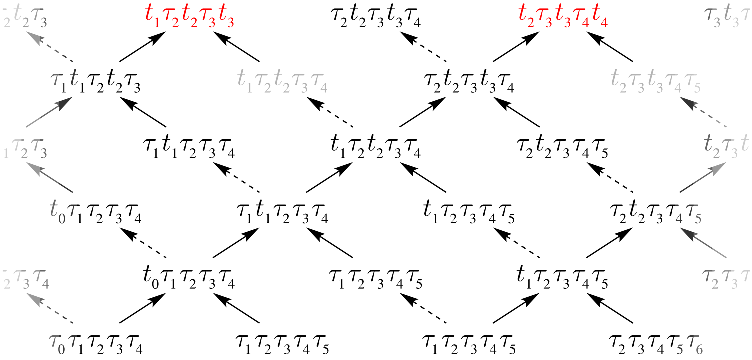


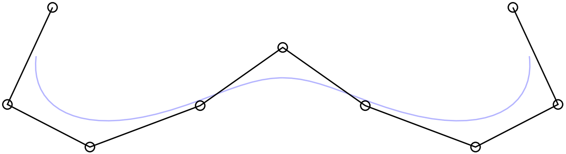

$\tau_1 t_1 \tau_2 t_2 \tau_3$
$t_1 \tau_2 t_2 \tau_3 \tau_4$
$\tau_2 t_2 \tau_3 t_3 \tau_4$
$t_2 \tau_3 t_3 \tau_4 \tau_5$
$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 t_2 \tau_3 \tau_4$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 \tau_3 \tau_4 \tau_5 \tau_6$

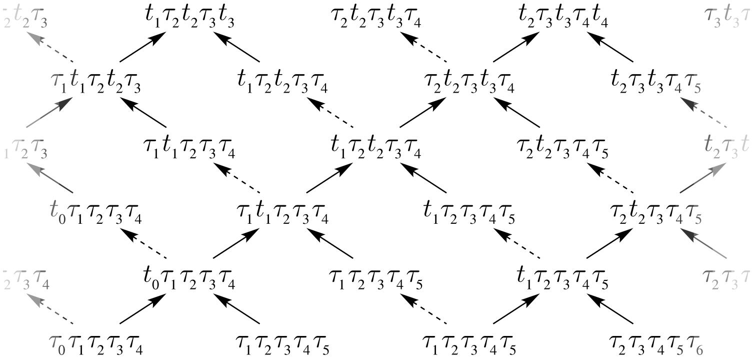



$\tau_1 t_1 \tau_2 t_2 \tau_3$
$t_1 \tau_2 t_2 \tau_3 \tau_4$
$\tau_2 t_2 \tau_3 t_3 \tau_4$
$t_2 \tau_3 t_3 \tau_4 \tau_5$
$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 t_2 \tau_3 \tau_4$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 t_1 \tau_2 \tau_3 \tau_4$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 t_2 \tau_3 \tau_4 \tau_5$
$t_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$t_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_0 \tau_1 \tau_2 \tau_3 \tau_4$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_1 \tau_2 \tau_3 \tau_4 \tau_5$
$\tau_2 \tau_3 \tau_4 \tau_5 \tau_6$



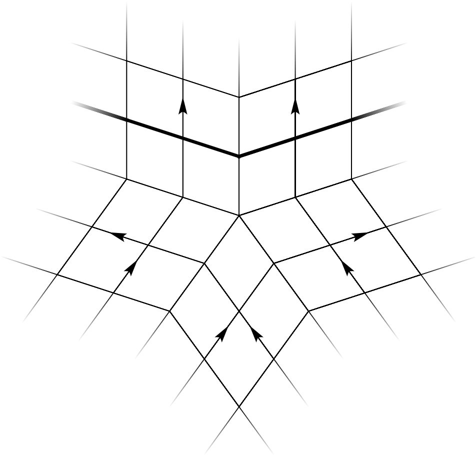

$t_1\tau_2t_2\tau_3t_3$
$\tau_2t_2\tau_3t_3\tau_4$
$t_2\tau_3t_3\tau_4t_4$
$\tau_1t_1\tau_2t_2\tau_3$
$t_1\tau_2t_2\tau_3\tau_4$
$\tau_2t_2\tau_3t_3\tau_4$
$t_2\tau_3t_3\tau_4\tau_5$
$\tau_1t_1\tau_2\tau_3\tau_4$
$t_1\tau_2t_2\tau_3\tau_4$
$\tau_2t_2\tau_3\tau_4\tau_5$
$t_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1t_1\tau_2\tau_3\tau_4$
$t_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2t_2\tau_3\tau_4\tau_5$
$t_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$t_1\tau_2\tau_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$




$t_1\tau_2t_2\tau_3t_3$
$\tau_2t_2\tau_3t_3\tau_4$
$t_2\tau_3t_3\tau_4t_4$
$\tau_1t_1\tau_2t_2\tau_3$
$t_1\tau_2t_2\tau_3\tau_4$
$\tau_2t_2\tau_3t_3\tau_4$
$t_2\tau_3t_3\tau_4\tau_5$
$\tau_1t_1\tau_2\tau_3\tau_4$
$t_1\tau_2t_2\tau_3\tau_4$
$\tau_2t_2\tau_3\tau_4\tau_5$
$t_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1t_1\tau_2\tau_3\tau_4$
$t_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2t_2\tau_3\tau_4\tau_5$
$t_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$t_1\tau_2\tau_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

Schaefer's algorithm is asymmetric

The asymmetry in Schaefer's algorithm makes it hard to use on surfaces



Revisiting our requirements

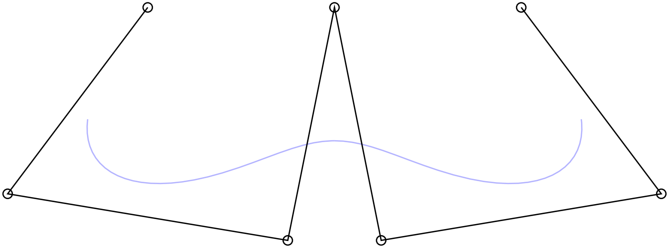
We want a non-uniform algorithm, where...

- Subdivision uses one refine and multiple smooth steps
- Smoothing filters compute a new point from two old points

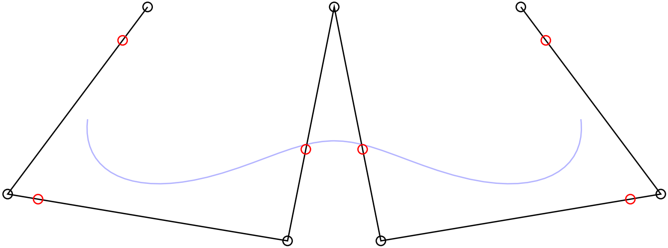
Revisiting our requirements

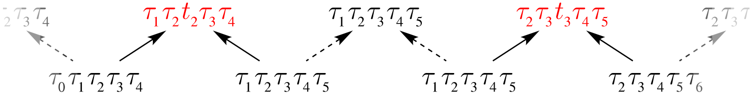
We want a non-uniform algorithm, where...

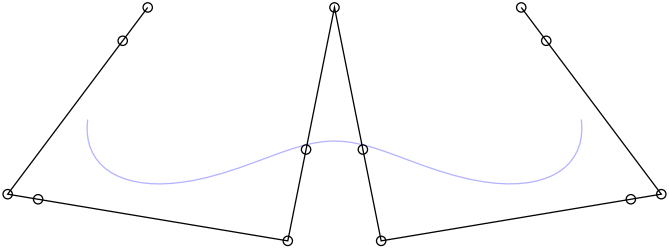
- Subdivision uses one refine and multiple smooth steps
- Smoothing filters compute a new point from two old points
- Intermediate smoothing steps are symmetric

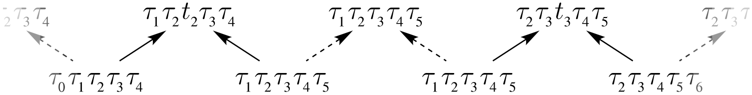


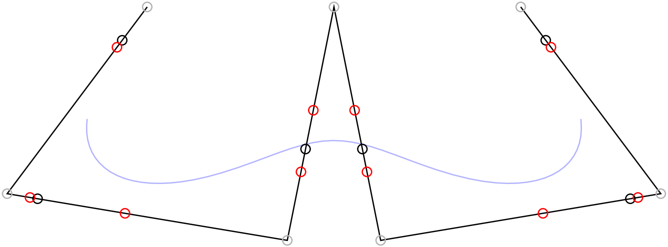
$\mathcal{T}_0\,\mathcal{T}_1\,\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4$ $\quad$ $\mathcal{T}_1\,\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4\,\mathcal{T}_5$ $\quad$ $\mathcal{T}_1\,\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4\,\mathcal{T}_5$ $\quad$ $\mathcal{T}_2\,\mathcal{T}_3\,\mathcal{T}_4\,\mathcal{T}_5\,\mathcal{T}_6$

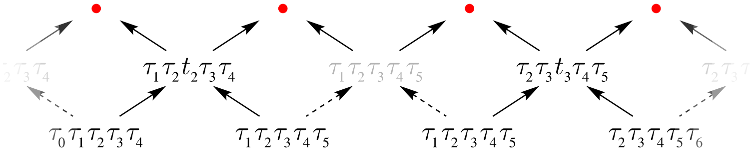


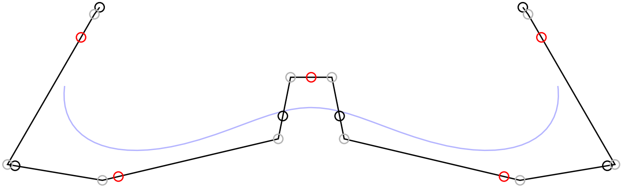

$\tau_3\tau_4$
$\tau_1\tau_2 t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3 t_3\tau_4\tau_5$
$\tau_2\tau_3$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

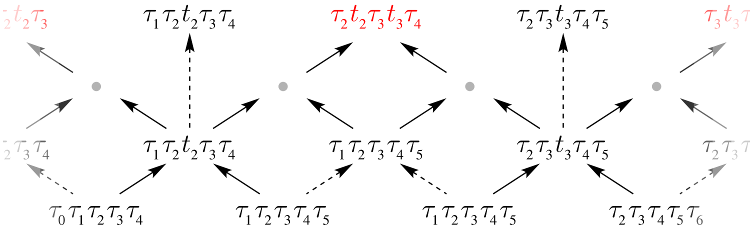


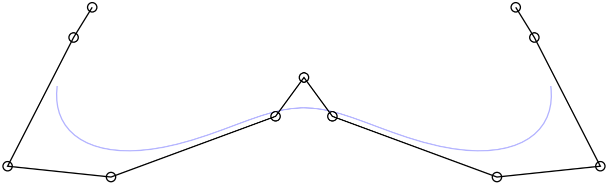

$\tau_1\tau_2 t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3 t_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

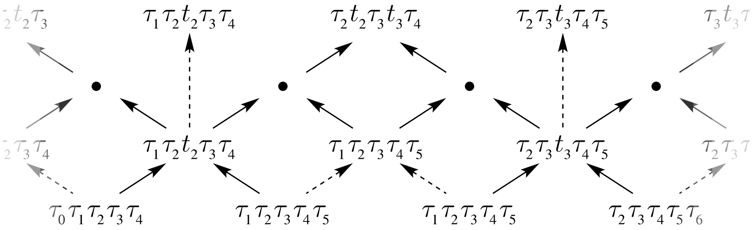


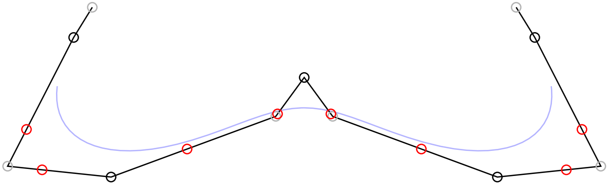

$\tau_2\tau_3\tau_4$
$\tau_1\tau_2 t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3 t_3\tau_4\tau_5$
$\tau_2\tau_3\tau$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

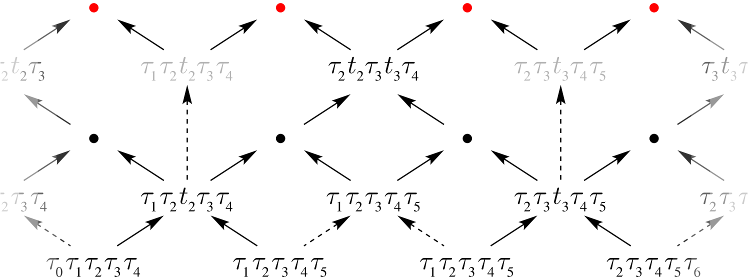


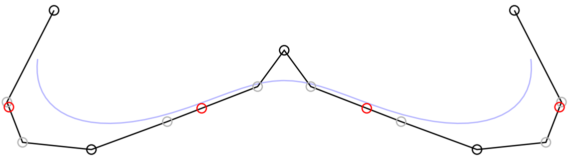

$\tau_1\tau_2 t_2\tau_3\tau_4$
$\tau_2 t_2\tau_3 t_3\tau_4$
$\tau_2\tau_3 t_3\tau_4\tau_5$
$\tau_1\tau_2 t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3 t_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

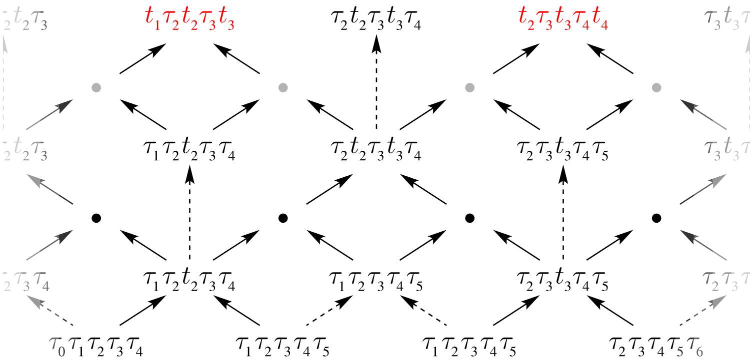


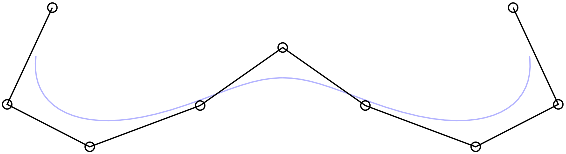

$\tau_1\tau_2t_2\tau_3\tau_4$
$\tau_2t_2\tau_3t_3\tau_4$
$\tau_2\tau_3t_3\tau_4\tau_5$
$\tau_1\tau_2t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3t_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

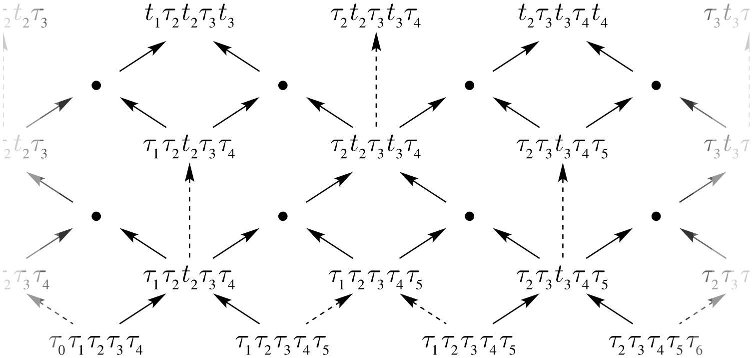



$\tau_1\tau_2t_2\tau_3\tau_4$
$\tau_2t_2\tau_3t_3\tau_4$
$\tau_2\tau_3t_3\tau_4\tau_5$
$\tau_1\tau_2t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3t_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

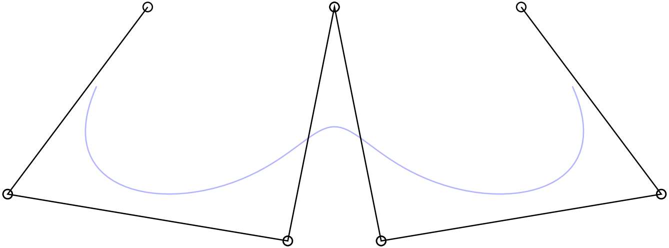



$t_1\tau_2t_2\tau_3t_3$
$\tau_2t_2\tau_3t_3\tau_4$
$t_2\tau_3t_3\tau_4t_4$
$\tau_1\tau_2t_2\tau_3\tau_4$
$\tau_2t_2\tau_3t_3\tau_4$
$\tau_2\tau_3t_3\tau_4\tau_5$
$\tau_1\tau_2t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3t_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$



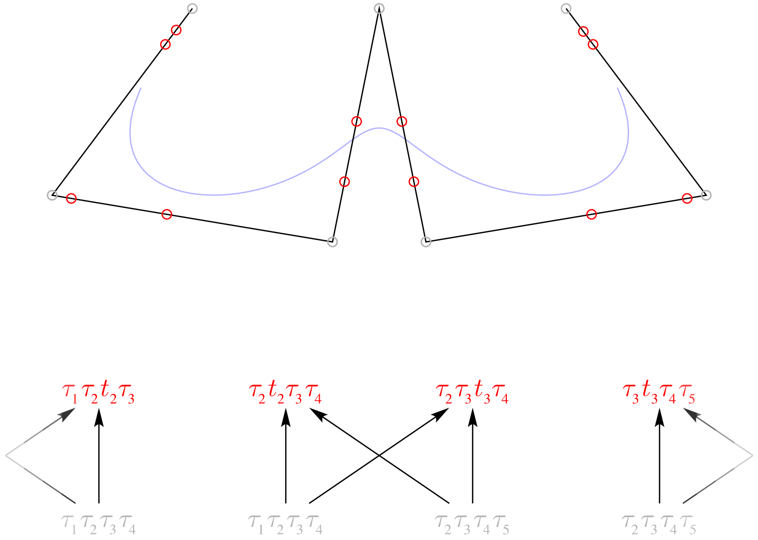

$t_1\tau_2 t_2\tau_3 t_3$
$\tau_2 t_2\tau_3 t_3\tau_4$
$t_2\tau_3 t_3\tau_4 t_4$
$\tau_1\tau_2 t_2\tau_3\tau_4$
$\tau_2 t_2\tau_3 t_3\tau_4$
$\tau_2\tau_3 t_3\tau_4\tau_5$
$\tau_1\tau_2 t_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3 t_3\tau_4\tau_5$
$\tau_0\tau_1\tau_2\tau_3\tau_4$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_1\tau_2\tau_3\tau_4\tau_5$
$\tau_2\tau_3\tau_4\tau_5\tau_6$

Even degree. . . and multiple knots

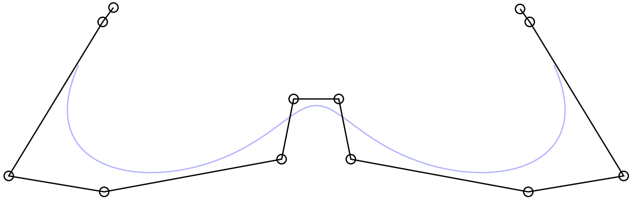


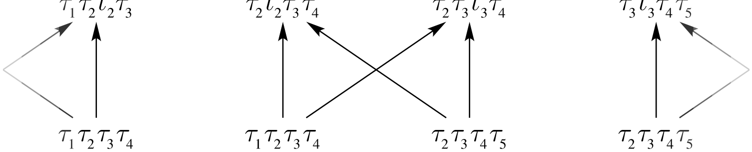
$\tau_1 \tau_2 \tau_3 \tau_4$ $\tau_1 \tau_2 \tau_3 \tau_4$ $\tau_2 \tau_3 \tau_4 \tau_5$ $\tau_2 \tau_3 \tau_4 \tau_5$

Even degree. . . and multiple knots

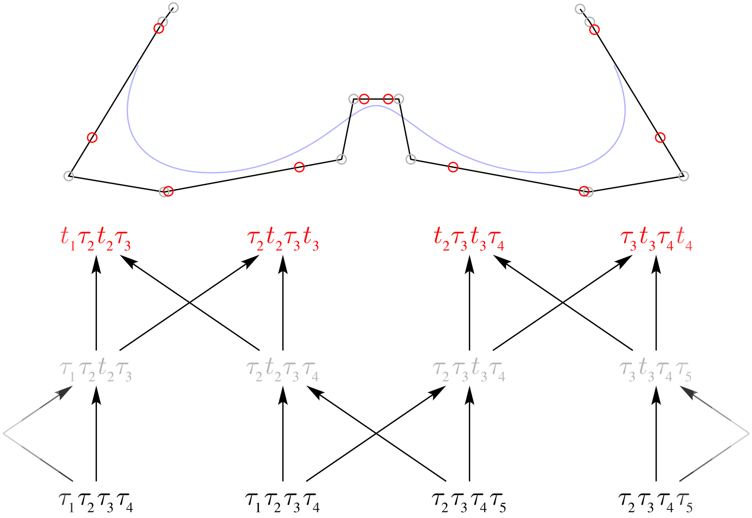


Even degree. . . and multiple knots

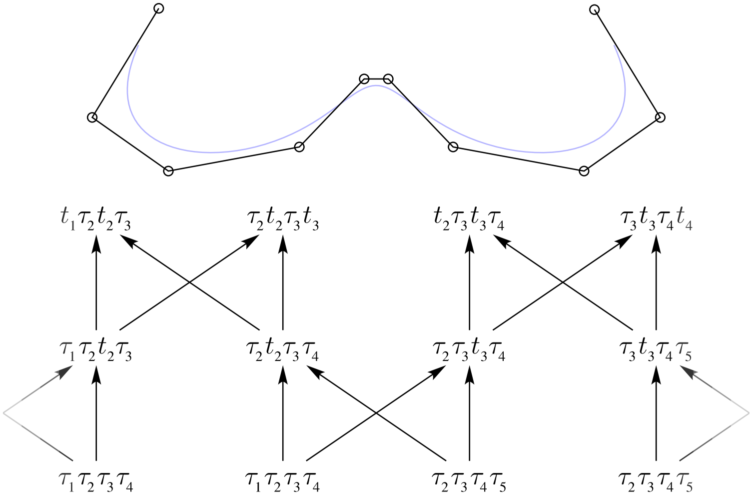




Even degree. . . and multiple knots



Even degree. . . and multiple knots

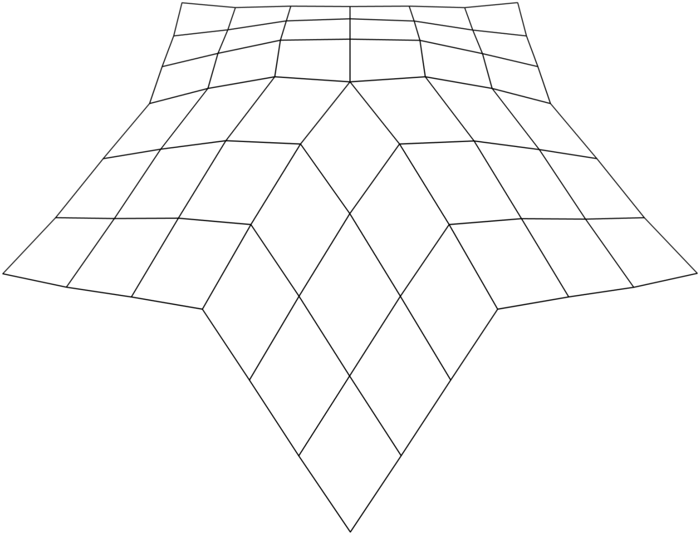


Summary

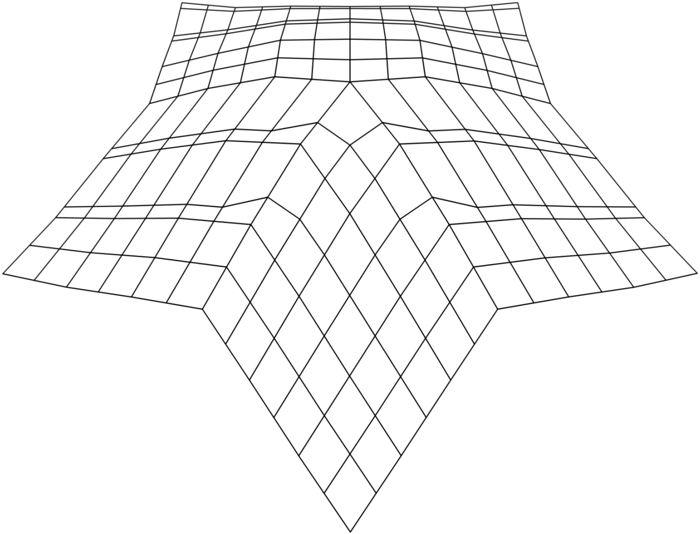
- There are non-uniform analogues of the Lane-Riesenfeld refine and smooth algorithm
- In fact there are several, each for different requirements
- A symmetric algorithm may lead to subdivision schemes generalising NURBS
- Multiple knots fit into a common framework

- Next step: extraordinary points

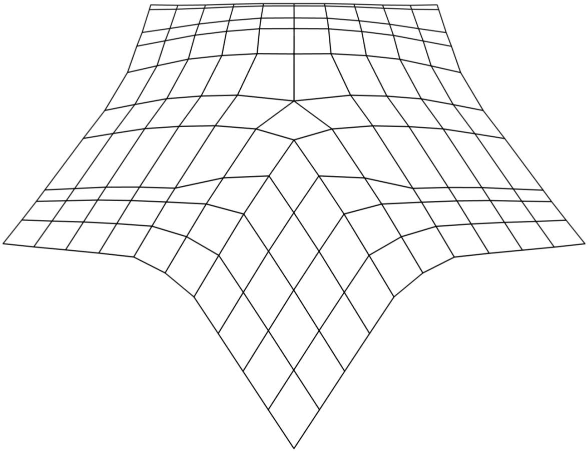
Extraordinary points



Extraordinary points



Extraordinary points



Extraordinary points