Geometric Modeling
(SS 2025)

Martin Held

FB Informatik
Universität Salzburg
A-5020 Salzburg, Austria
held@cs.sbg.ac.at

June 13, 2025

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Personalia

Instructor (VO+PS): M. Held.

Email: held@cs.sbg.ac.at.

Base-URL: https://www.cosy.sbg.ac.at/~held.

Office: Universität Salzburg, FB Informatik, Rm. 1.20,
Jakob-Haringer Str. 2, 5020 Salzburg-Itzling.

Phone number (office): (0662) 8044-6304.

Phone number (secr.): (0662) 8044-6300.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Formalia

URL of course (VO+PS): Base-URL/teaching/geom_mod/geom_mod.html.

Lecture times (VO): Friday 9^{10}–11^{10}.

Venue (VO): T03, PLUS, FB Informatik, Jakob-Haringer Str. 2.

Lecture times (PS): Friday 7^{45}–8^{50}.

Venue (PS): T03, PLUS, FB Informatik, Jakob-Haringer Str. 2.

Note — PS is graded according to continuous-assessment mode!

Electronic Slides and Online Material

In addition to these slides, you are encouraged to consult the WWW home-page of this lecture:

https://www.cosy.sbg.ac.at/~held/teaching/geom_mod/geom_mod.html.

In particular, this WWW page contains up-to-date information on the course, plus links to online notes, slides and (possibly) sample code.

A Few Words of Warning

- I hope that these slides will serve as a practice-minded introduction to various aspects of geometric modeling. I would like to warn you explicitly not to regard these slides as the sole source of information on the topics of my course. It may and will happen that I'll use the lecture for talking about subtle details that need not be covered in these slides! In particular, the slides won't contain all sample calculations, proofs of theorems, demonstrations of algorithms, or solutions to problems posed during my lecture. That is, by making these slides available to you I do not intend to encourage you to attend the lecture on an irregular basis.
- See also In Praise of Lectures by T.W. Körner.
- A *basic knowledge of calculus, linear algebra, discrete mathematics, and geometric computing*, as taught in standard undergraduate CS courses, should suffice to take this course. It is my sincere intention to start at such a hypothetical low level of "typical prior undergrad knowledge". Still, it is obvious that different educational backgrounds will result in different levels of prior knowledge. Hence, you might realize that you do already know some items covered in this course, while you lack a decent understanding of some other items which I seem to presuppose. In such a case I do expect you to refresh or fill in those missing items on your own!

Acknowledgments

A small portion of these slides is based on notes and slides originally prepared by students — most notably Dominik Kaaser, Kamran Safdar, and Marko Šulejić — on topics related to geometric modeling. I would like to express my thankfulness to all of them for their help. This revision and extension was carried out by myself, and I am responsible for all errors.

I am also happy to acknowledge that I benefited from material published by colleagues on diverse topics that are partially covered in this lecture. While some of the material used for this lecture was originally presented in traditional-style publications (such as textbooks), some other material has its roots in non-standard publication outlets (such as online documentations, electronic course notes, or user manuals).

Salzburg, February 2025 Martin Held

Legal Fine Print and Disclaimer

To the best of my knowledge, these slides do not violate or infringe upon somebody else's copyrights. If copyrighted material appears in these slides then it was considered to be available in a non-profit manner and as an educational tool for teaching at an academic institution, within the limits of the "fair use" policy. For copyrighted material we strive to give references to the copyright holders (if known). Of course, any trademarks mentioned in these slides are properties of their respective owners.

Please note that these slides are copyrighted. The copyright holder grants you the right to download and print the slides for your personal use. Any other use, including instructional use at non-profit academic institutions and re-distribution in electronic or printed form of significant portions, beyond the limits of "fair use", requires the explicit permission of the copyright holder. All rights reserved.

These slides are made available without warrant of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. In no event shall the copyright holder and/or his respective employer be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, arising out of or in connection with the use of information provided in these slides.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Recommended Textbooks I

- G. Farin.
  Curves and Surfaces for CAGD: A Practical Guide.
  Morgan Kaufmann, 5th edition, 2002; ISBN 978-1-55860-737-8.
- H. Prautzsch, W. Boehm, M. Paluszny.
  Bézier and B-spline Techniques.
  Springer, 2002; ISBN 978-3540437611.
  `https://link.springer.com/book/10.1007/978-3-662-04919-8`
- J. Gallier.
  Curves and Surfaces in Geometric Modeling.
  Morgan Kaufmann, 1999; ISBN 978-1558605992.
  `http://www.cis.upenn.edu/~jean/gbooks/geom1.html`
- R. Goldman.
  An Integrated Introduction to Computer Graphics and Geometric Modeling.
  CRC Press, 2019; ISBN 978-1-138-38147-6.

Recommended Textbooks II

- N.M. Patrikalakis, T. Maekawa, W. Cho.
  Shape Interrogation for Computer Aided Design and Manufacturing.
  Springer, 2nd corr. edition, 2010; ISBN 978-3642040733.
  http://web.mit.edu/hyperbook/Patrikalakis-Maekawa-Cho/
- M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Levy.
  Polygon Mesh Processing.
  A K Peters/CRC Press, 2010; ISBN 978-1568814261.
  http://www.pmp-book.org/
- A. Dickenstein, I.Z. Emiris (eds.).
  Solving Polynomial Equations: Foundations, Algorithms, and Applications.
  Springer, 2005; ISBN 978-3-540-27357-8.
- G.E. Farin, D. Hansford.
  Practical Linear Algebra: A Geometry Toolbox.
  A K Peters/CRC Press, 4th edition, 2021; ISBN 978-0367507848.

Table of Content

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

1 Introduction

- Motivation
- Notation

Motivation: Evaluation of a Polynomial

- Assume that we have an intuitive understanding of polynomials and consider a polynomial in x of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: Evaluation of a Polynomial

- Assume that we have an intuitive understanding of polynomials and consider a polynomial in x of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

- A straightforward polynomial evaluation of p for a given parameter x_0 — i.e., the computation of $p(x_0)$ — results in k multiplications for a monomial of degree k, plus a total of n additions.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: Evaluation of a Polynomial

- Assume that we have an intuitive understanding of polynomials and consider a polynomial in x of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

- A straightforward polynomial evaluation of p for a given parameter x_0 — i.e., the computation of $p(x_0)$ — results in k multiplications for a monomial of degree k, plus a total of n additions.
- Hence, we would get

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2} = O(n^2)$$

multiplications (and n additions).

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: Evaluation of a Polynomial

- Assume that we have an intuitive understanding of polynomials and consider a polynomial in x of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

- A straightforward polynomial evaluation of p for a given parameter x_0 — i.e., the computation of $p(x_0)$ — results in k multiplications for a monomial of degree k, plus a total of n additions.
- Hence, we would get

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2} = O(n^2)$$

multiplications (and n additions).
- Can we do better?

Motivation: Evaluation of a Polynomial

- Assume that we have an intuitive understanding of polynomials and consider a polynomial in x of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

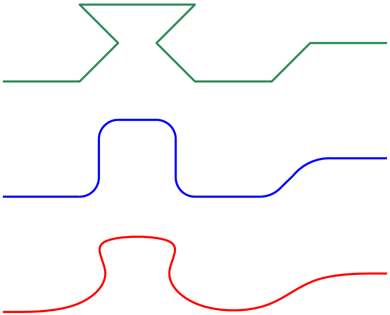
- A straightforward polynomial evaluation of p for a given parameter x_0 — i.e., the computation of $p(x_0)$ — results in k multiplications for a monomial of degree k, plus a total of n additions.
- Hence, we would get

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2} = O(n^2)$$

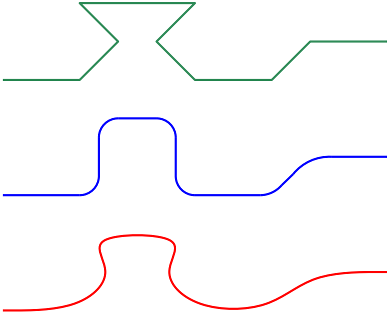
  multiplications (and n additions).
- Can we do better?
- Yes, we can: Horner's Algorithm consumes only n multiplications and n additions to evaluate a polynomial of degree n!

Motivation: Smoothness of a Curve

- What is a characteristic difference between the three curves shown below?



- What is a characteristic difference between the three curves shown below?

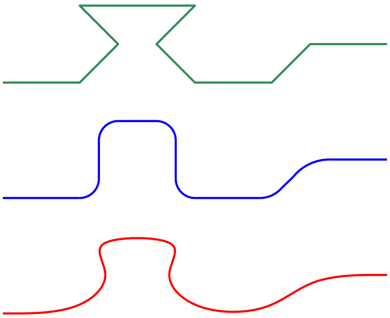


- Answer: The green curve has tangential discontinuities at the vertices, the blue curve consists of straight-line segments and circular arcs and is tangent-continuous, while the red curve is a cubic B-spline and is curvature-continuous.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: Smoothness of a Curve

- What is a characteristic difference between the three curves shown below?

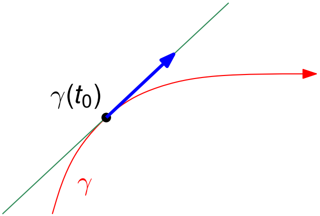


- Answer: The green curve has tangential discontinuities at the vertices, the blue curve consists of straight-line segments and circular arcs and is tangent-continuous, while the red curve is a cubic B-spline and is curvature-continuous.
- By the way, when precisely is a geometric object a "curve"?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

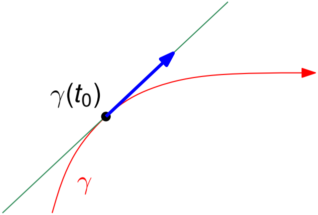
Motivation: Tangent to a Curve

- What is a parametrization of the tangent line at a point $\gamma(t_0)$ of a curve γ?



Motivation: Tangent to a Curve

- What is a parametrization of the tangent line at a point $\gamma(t_0)$ of a curve γ?

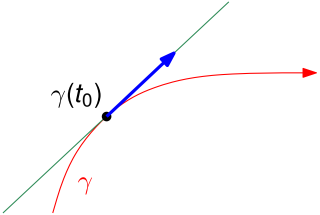


- Answer: If γ is differentiable then a parametrization of the tangent line ℓ that passes through $\gamma(t_0)$ is given by

$$\ell(\lambda) = \gamma(t_0) + \lambda\gamma'(t_0) \quad \text{with } \lambda \in \mathbb{R}.$$

Motivation: Tangent to a Curve

- What is a parametrization of the tangent line at a point $\gamma(t_0)$ of a curve γ?



- Answer: If γ is differentiable then a parametrization of the tangent line ℓ that passes through $\gamma(t_0)$ is given by

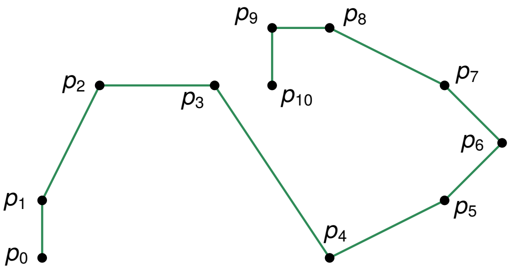
$$\ell(\lambda) = \gamma(t_0) + \lambda\gamma'(t_0) \quad \text{with } \lambda \in \mathbb{R}.$$

- How can we obtain $\gamma'(t)$ for $\gamma\colon \mathbb{R} \to \mathbb{R}^d$?

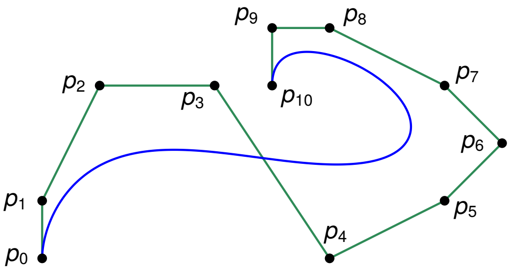
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: Bézier Curve

- How can we model a "smooth" polynomial curve in $\mathbb{R}^2$ by specifying a sequence of so-called "control points". (E.g., the points $p_0, p_1, \ldots, p_{10}$ in the figure.)



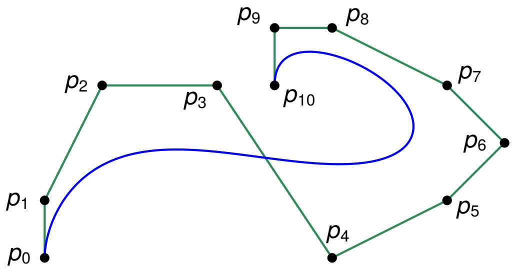
- How can we model a "smooth" polynomial curve in $\mathbb{R}^2$ by specifying a sequence of so-called "control points". (E.g., the points $p_0, p_1, \ldots, p_{10}$ in the figure.)



- One (widely used) option is to generate a Bézier curve. (The figure shows a Bézier curve of degree 10 with 11 control points.)

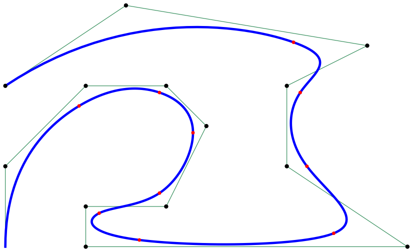
Motivation: Bézier Curve

- How can we model a “smooth” polynomial curve in $\mathbb{R}^2$ by specifying a sequence of so-called “control points”. (E.g., the points $p_0, p_1, \ldots, p_{10}$ in the figure.)



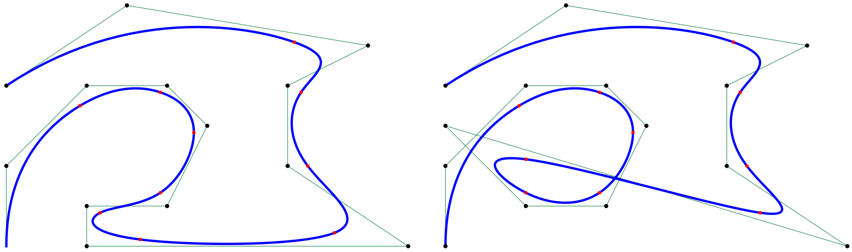
- One (widely used) option is to generate a Bézier curve. (The figure shows a Bézier curve of degree 10 with 11 control points.)
- What is the degree of a Bézier curve? Which geometric and mathematical properties do Bézier curves exhibit?

- How can we model a (piecewise) polynomial curve in $\mathbb{R}^2$ by specifying a sequence of so-called “control points” such that a modification of one control point affects only a “small” portion of the curve?



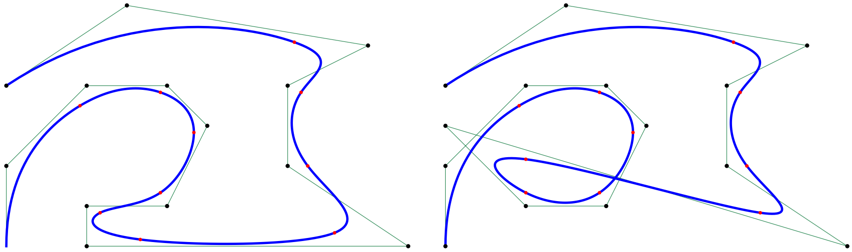
Motivation: B-Spline Curve

- How can we model a (piecewise) polynomial curve in $\mathbb{R}^2$ by specifying a sequence of so-called "control points" such that a modification of one control point affects only a "small" portion of the curve?



Motivation: B-Spline Curve

- How can we model a (piecewise) polynomial curve in $\mathbb{R}^2$ by specifying a sequence of so-called “control points” such that a modification of one control point affects only a “small” portion of the curve?



- Answer: Use B-spline curves.
- Which geometric and mathematical properties do B-spline curves exhibit?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: NURBS

- Is it possible to parameterize a circular arc by means of a polynomial term? Or by a rational term?

- Is it possible to parameterize a circular arc by means of a polynomial term? Or by a rational term?
- Yes, this is possible by means of a rational term:

$$\left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right) \qquad \text{for } t \in \mathbb{R}.$$

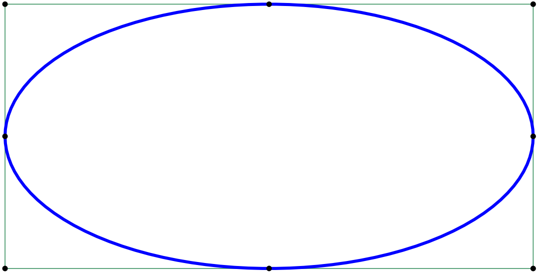
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: NURBS

- Is it possible to parameterize a circular arc by means of a polynomial term? Or by a rational term?
- Yes, this is possible by means of a rational term:

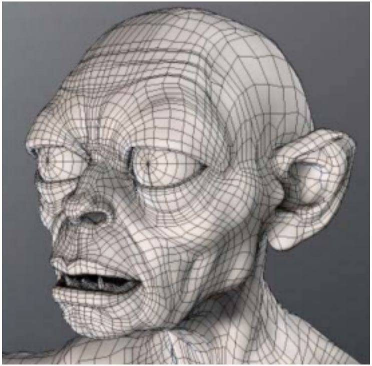
$$\left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right) \qquad \text{for } t \in \mathbb{R}.$$

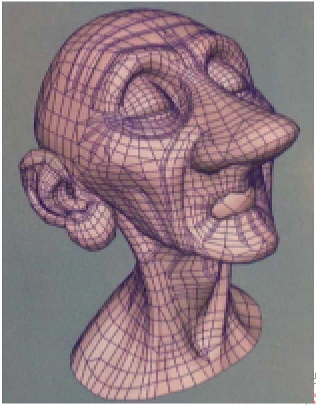
- More generally, NURBS can be used to model all types of conics by means of rational parametrizations.



Motivation: Modeling Complicated Organic Shapes

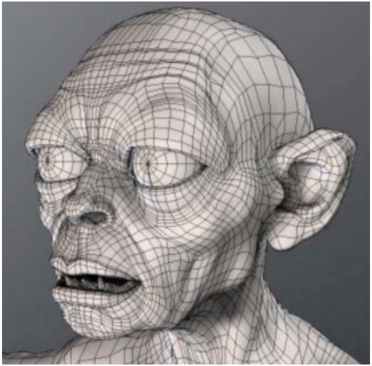
- How can we model a complicated organic shape such as (humanoid) characters like Gollum (from “Lord of the Rings”), or Geri (from Pixar’s “Geri’s Game”)?

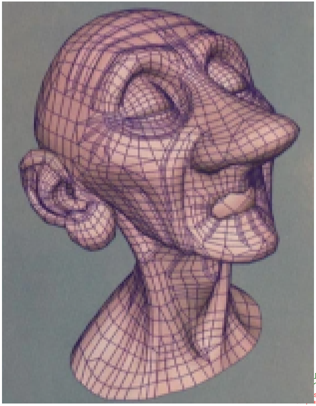




Motivation: Modeling Complicated Organic Shapes

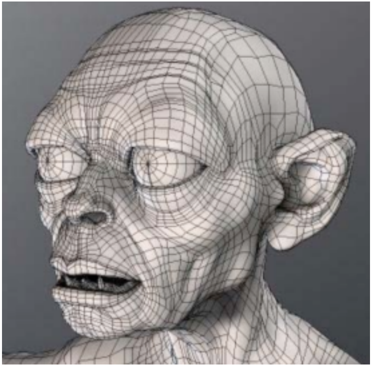
- How can we model a complicated organic shape such as (humanoid) characters like Gollum (from “Lord of the Rings”), or Geri (from Pixar’s “Geri’s Game”)?
- In theory, spline-based modeling is possible.

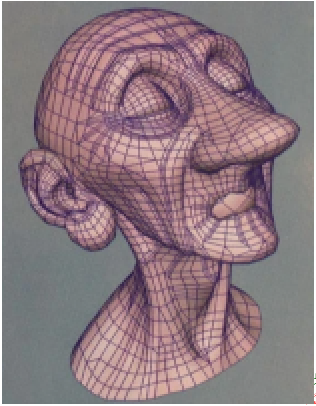




Motivation: Modeling Complicated Organic Shapes

- How can we model a complicated organic shape such as (humanoid) characters like Gollum (from "Lord of the Rings"), or Geri (from Pixar's "Geri's Game")?
- In theory, spline-based modeling is possible.
- In practice, subdivision surfaces are easier to deal with.





Motivation: Approximation of a Continuous Function

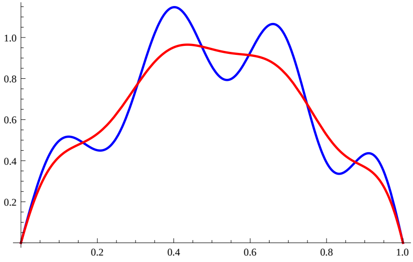
- How can we approximate a continuous function by a polynomial?

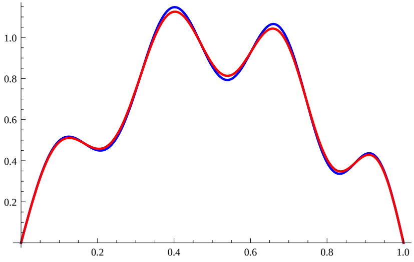
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation: Approximation of a Continuous Function

- How can we approximate a continuous function by a polynomial?
- Answer: We can use a Bernstein approximation.
- Sample Bernstein approximations of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad\qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$

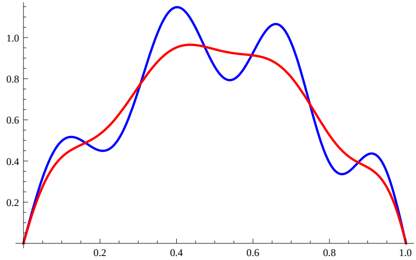


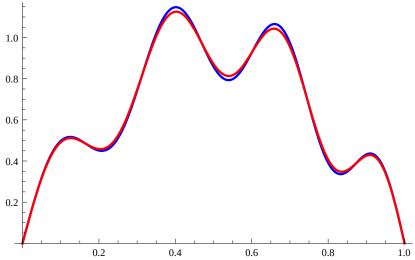


Motivation: Approximation of a Continuous Function

- How can we approximate a continuous function by a polynomial?
- Answer: We can use a Bernstein approximation.
- Sample Bernstein approximations of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad\qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$





- One can prove that the Bernstein approximation $B_{n,f}$ converges uniformly to f on the interval $[0, 1]$ as n increases, for every continuous function f.

Notation: Numbers and Sets

- Numbers:
  - The set $\{1, 2, 3, \ldots\}$ of natural numbers is denoted by $\mathbb{N}$, with $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.
  - The set $\{2, 3, 5, 7, 11, 13, \ldots\} \subset \mathbb{N}$ of prime numbers is denoted by $\mathbb{P}$.
  - The (positive and negative) integers are denoted by $\mathbb{Z}$.
  - $\mathbb{Z}_n := \{0, 1, 2, \ldots, n-1\}$ and $\mathbb{Z}_n^+ := \{1, 2, \ldots, n-1\}$ for $n \in \mathbb{N}$.
  - The reals are denoted by $\mathbb{R}$; the non-negative reals are denoted by $\mathbb{R}_0^+$, and the positive reals by $\mathbb{R}^+$.

Notation: Numbers and Sets

- Numbers:
  - The set $\{1, 2, 3, \ldots\}$ of natural numbers is denoted by $\mathbb{N}$, with $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.
  - The set $\{2, 3, 5, 7, 11, 13, \ldots\} \subset \mathbb{N}$ of prime numbers is denoted by $\mathbb{P}$.
  - The (positive and negative) integers are denoted by $\mathbb{Z}$.
  - $\mathbb{Z}_n := \{0, 1, 2, \ldots, n-1\}$ and $\mathbb{Z}_n^+ := \{1, 2, \ldots, n-1\}$ for $n \in \mathbb{N}$.
  - The reals are denoted by $\mathbb{R}$; the non-negative reals are denoted by $\mathbb{R}_0^+$, and the positive reals by $\mathbb{R}^+$.
- Open or closed intervals $I \subset \mathbb{R}$ are denoted using square brackets: e.g., $I_1 = [a_1, b_1]$ or $I_2 = [a_2, b_2[$, with $a_1, a_2, b_1, b_2 \in \mathbb{R}$, where the right-hand "[" indicates that the value b_2 is not included in I_2.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Notation: Numbers and Sets

- Numbers:
  - The set $\{1, 2, 3, \dots\}$ of natural numbers is denoted by $\mathbb{N}$, with $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.
  - The set $\{2, 3, 5, 7, 11, 13, \dots\} \subset \mathbb{N}$ of prime numbers is denoted by $\mathbb{P}$.
  - The (positive and negative) integers are denoted by $\mathbb{Z}$.
  - $\mathbb{Z}_n := \{0, 1, 2, \dots, n-1\}$ and $\mathbb{Z}_n^+ := \{1, 2, \dots, n-1\}$ for $n \in \mathbb{N}$.
  - The reals are denoted by $\mathbb{R}$; the non-negative reals are denoted by $\mathbb{R}_0^+$, and the positive reals by $\mathbb{R}^+$.
- Open or closed intervals $I \subset \mathbb{R}$ are denoted using square brackets: e.g., $I_1 = [a_1, b_1]$ or $I_2 = [a_2, b_2[$, with $a_1, a_2, b_1, b_2 \in \mathbb{R}$, where the right-hand "[" indicates that the value b_2 is not included in I_2.
- The set of all elements $a \in A$ with property $P(a)$, for some set A and some predicate P, is denoted by

$$\{x \in A : \; P(x)\} \quad \text{or} \quad \{x : \; x \in A \wedge \; P(x)\}$$

or

$$\{x \in A \mid P(x)\} \quad \text{or} \quad \{x \mid x \in A \wedge \; P(x)\}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Notation: Numbers and Sets

- Numbers:
  - The set $\{1, 2, 3, \ldots\}$ of natural numbers is denoted by $\mathbb{N}$, with $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.
  - The set $\{2, 3, 5, 7, 11, 13, \ldots\} \subset \mathbb{N}$ of prime numbers is denoted by $\mathbb{P}$.
  - The (positive and negative) integers are denoted by $\mathbb{Z}$.
  - $\mathbb{Z}_n := \{0, 1, 2, \ldots, n-1\}$ and $\mathbb{Z}_n^+ := \{1, 2, \ldots, n-1\}$ for $n \in \mathbb{N}$.
  - The reals are denoted by $\mathbb{R}$; the non-negative reals are denoted by $\mathbb{R}_0^+$, and the positive reals by $\mathbb{R}^+$.
- Open or closed intervals $I \subset \mathbb{R}$ are denoted using square brackets: e.g., $I_1 = [a_1, b_1]$ or $I_2 = [a_2, b_2[$, with $a_1, a_2, b_1, b_2 \in \mathbb{R}$, where the right-hand "[" indicates that the value b_2 is not included in I_2.
- The set of all elements $a \in A$ with property $P(a)$, for some set A and some predicate P, is denoted by

$$\{x \in A : \; P(x)\} \quad \text{or} \quad \{x : \; x \in A \wedge \; P(x)\}$$

or

$$\{x \in A \mid P(x)\} \quad \text{or} \quad \{x \mid x \in A \wedge \; P(x)\}.$$

- Quantifiers: The universal quantifier is denoted by $\forall$, and $\exists$ denotes the existential quantifier.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Notation: Numbers and Sets

- Numbers:
  - The set $\{1, 2, 3, \ldots\}$ of natural numbers is denoted by $\mathbb{N}$, with $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.
  - The set $\{2, 3, 5, 7, 11, 13, \ldots\} \subset \mathbb{N}$ of prime numbers is denoted by $\mathbb{P}$.
  - The (positive and negative) integers are denoted by $\mathbb{Z}$.
  - $\mathbb{Z}_n := \{0, 1, 2, \ldots, n-1\}$ and $\mathbb{Z}_n^+ := \{1, 2, \ldots, n-1\}$ for $n \in \mathbb{N}$.
  - The reals are denoted by $\mathbb{R}$; the non-negative reals are denoted by $\mathbb{R}_0^+$, and the positive reals by $\mathbb{R}^+$.
- Open or closed intervals $I \subset \mathbb{R}$ are denoted using square brackets: e.g., $I_1 = [a_1, b_1]$ or $I_2 = [a_2, b_2[$, with $a_1, a_2, b_1, b_2 \in \mathbb{R}$, where the right-hand "[" indicates that the value b_2 is not included in I_2.
- The set of all elements $a \in A$ with property $P(a)$, for some set A and some predicate P, is denoted by

  $$\{x \in A : \; P(x)\} \quad \text{or} \quad \{x : \; x \in A \wedge \; P(x)\}$$

  or

  $$\{x \in A \mid P(x)\} \quad \text{or} \quad \{x \mid x \in A \wedge \; P(x)\}.$$

- Quantifiers: The universal quantifier is denoted by $\forall$, and $\exists$ denotes the existential quantifier.
- Bold capital letters, such as **M**, are used for matrices.
- The set of all (real) $m \times n$ matrices is denoted by $M_{m \times n}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Notation: Vectors

- Points are denoted by letters written in italics: p, q or, occasionally, P, Q. We do not distinguish between a point and its position vector.
- The coordinates of a vector are denoted by using indices (or numbers): e.g., $v = (v_x, v_y)$ for $v \in \mathbb{R}^2$, or $v = (v_1, v_2, \ldots, v_n)$ for $v \in \mathbb{R}^n$.
- In order to state $v \in \mathbb{R}^n$ in vector form we will mix column and row vectors freely unless a specific form is required, such as for matrix multiplication.

Notation: Vectors

- Points are denoted by letters written in italics: p, q or, occasionally, P, Q. We do not distinguish between a point and its position vector.
- The coordinates of a vector are denoted by using indices (or numbers): e.g., $v = (v_x, v_y)$ for $v \in \mathbb{R}^2$, or $v = (v_1, v_2, \ldots, v_n)$ for $v \in \mathbb{R}^n$.
- In order to state $v \in \mathbb{R}^n$ in vector form we will mix column and row vectors freely unless a specific form is required, such as for matrix multiplication.
- The vector dot product of two vectors $v, w \in \mathbb{R}^n$ is denoted by $\langle v, w \rangle$. That is, $\langle v, w \rangle = \sum_{i=1}^{n} v_i \cdot w_i$ for $v, w \in \mathbb{R}^n$.
- The vector cross-product (in $\mathbb{R}^3$) is denoted by a cross: $v \times w$.
- The length of a vector v is denoted by $\|v\|$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Notation: Vectors

- Points are denoted by letters written in italics: p, q or, occasionally, P, Q. We do not distinguish between a point and its position vector.
- The coordinates of a vector are denoted by using indices (or numbers): e.g., $v = (v_x, v_y)$ for $v \in \mathbb{R}^2$, or $v = (v_1, v_2, \ldots, v_n)$ for $v \in \mathbb{R}^n$.
- In order to state $v \in \mathbb{R}^n$ in vector form we will mix column and row vectors freely unless a specific form is required, such as for matrix multiplication.
- The vector dot product of two vectors $v, w \in \mathbb{R}^n$ is denoted by $\langle v, w \rangle$. That is, $\langle v, w \rangle = \sum_{i=1}^{n} v_i \cdot w_i$ for $v, w \in \mathbb{R}^n$.
- The vector cross-product (in $\mathbb{R}^3$) is denoted by a cross: $v \times w$.
- The length of a vector v is denoted by $\|v\|$.
- The straight-line segment between the points p and q is denoted by $\overline{pq}$.
- The supporting line of the points p and q is denoted by $\ell(p, q)$.

Notation: Sum and Product

- Consider k real numbers $a_1, a_2, \ldots, a_k \in \mathbb{R}$, together with some $m, n \in \mathbb{N}$ such that $1 \leq m, n \leq k$.

$$\sum_{i=m}^{n} a_i := \begin{cases} 0 & \text{if} \quad n < m, \\ a_m & \text{if} \quad n = m, \\ \left(\sum_{i=m}^{n-1} a_i\right) + a_n & \text{if} \quad n > m. \end{cases}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Notation: Sum and Product

- Consider k real numbers $a_1, a_2, \ldots, a_k \in \mathbb{R}$, together with some $m, n \in \mathbb{N}$ such that $1 \leq m, n \leq k$.

$$\sum_{i=m}^{n} a_i := \begin{cases} 0 & \text{if} \quad n < m, \\ a_m & \text{if} \quad n = m, \\ \left(\sum_{i=m}^{n-1} a_i\right) + a_n & \text{if} \quad n > m. \end{cases}$$

$$\prod_{i=m}^{n} a_i := \begin{cases} 1 & \text{if} \quad n < m, \\ a_m & \text{if} \quad n = m, \\ \left(\prod_{i=m}^{n-1} a_i\right) \cdot a_n & \text{if} \quad n > m. \end{cases}$$

2 Mathematics for Geometric Modeling

2 Mathematics for Geometric Modeling

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 1 (Least element, Dt.: kleinstes Element, Minimum)

Consider $T \subseteq \mathbb{R}$. An element $a \in T$ is a *least element* (or *minimum*) of T if $\forall b \in T \setminus \{a\} \quad a \leq b$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Extreme Elements

Definition 1 (Least element, Dt.: kleinstes Element, Minimum)

Consider $T \subseteq \mathbb{R}$. An element $a \in T$ is a *least element* (or *minimum*) of T if $\forall b \in T \setminus \{a\} \quad a \leq b$.

- This definition can be extended to an arbitrary partially-ordered set $(S, \preceq)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Extreme Elements

Definition 1 (Least element, Dt.: kleinstes Element, Minimum)

Consider $T \subseteq \mathbb{R}$. An element $a \in T$ is a *least element* (or *minimum*) of T if $\forall b \in T \setminus \{a\} \quad a \leq b$.

- This definition can be extended to an arbitrary partially-ordered set $(S, \preceq)$.

Definition 2 (Greatest element, Dt.: größtes Element, Maximum)

Consider $T \subseteq \mathbb{R}$. An element $a \in T$ is a *greatest element* (or *maximum*) of T if $\forall b \in T \setminus \{a\} \quad b \leq a$.

Definition 3 (Lower bound, Dt.: untere Schranke)

Consider $T \subseteq \mathbb{R}$. The set T is *bounded below* if there exists an element $s \in \mathbb{R}$, a *lower bound* of T, such that

$$\forall t \in T \quad s \leq t.$$

Infimum

Definition 3 (Lower bound, Dt.: untere Schranke)

Consider $T \subseteq \mathbb{R}$. The set T is *bounded below* if there exists an element $s \in \mathbb{R}$, a *lower bound* of T, such that

$$\forall t \in T \quad s \leq t.$$

Definition 4 (Greatest lower bound, infimum, Dt.: Infimum, größte untere Schranke)

Consider $T \subseteq \mathbb{R}$. An element $s \in \mathbb{R}$ is called *greatest lower bound* (or *infimum* of T), and denoted by $\inf(T)$, if

$$\forall t \in T \quad s \leq t \quad \text{and} \quad \forall s' \in \mathbb{R} \quad \big((\forall t \in T \;\; s' \leq t) \;\Rightarrow\; s' \leq s\big).$$

Infimum

Definition 3 (Lower bound, Dt.: untere Schranke)

Consider $T \subseteq \mathbb{R}$. The set T is *bounded below* if there exists an element $s \in \mathbb{R}$, a *lower bound* of T, such that

$$\forall t \in T \quad s \leq t.$$

Definition 4 (Greatest lower bound, infimum, Dt.: Infimum, größte untere Schranke)

Consider $T \subseteq \mathbb{R}$. An element $s \in \mathbb{R}$ is called *greatest lower bound* (or *infimum* of T), and denoted by $\inf(T)$, if

$$\forall t \in T \quad s \leq t \quad \text{and} \quad \forall s' \in \mathbb{R} \quad ((\forall t \in T \;\; s' \leq t) \;\Rightarrow\; s' \leq s).$$

Mind the difference!

The terms “minimum” and “infimum” have different meanings!

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 5

Consider $T \subseteq \mathbb{R}$.
(1) If the infimum of T exists then it is unique.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Infimum and Supremum

Lemma 5

Consider $T \subseteq \mathbb{R}$.
(1) If the infimum of T exists then it is unique.
(2) If the infimum of T belongs to T then it is also the minimum of T.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 5

Consider $T \subseteq \mathbb{R}$.
(1) If the infimum of T exists then it is unique.
(2) If the infimum of T belongs to T then it is also the minimum of T.

- The definitions of *upper bound* and *supremum* are obtained by replacing terms like “lower” by “upper” in these definitions.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

2 Mathematics for Geometric Modeling

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 6 (Factorial, Dt.: Fakultät, Faktorielle)

For $n \in \mathbb{N}_0$,

$$n! \;:=\; \begin{cases} 1 & \text{if } \; n \leq 1, \\ n \cdot (n-1)! & \text{if } \; n > 1. \end{cases}$$

Definition 6 (Factorial, Dt.: Fakultät, Faktorielle)

For $n \in \mathbb{N}_0$,

$$n! \;:=\; \begin{cases} 1 & \text{if } n \leq 1, \\ n \cdot (n-1)! & \text{if } n > 1. \end{cases}$$

- Note that $0! = 1$ by definition!

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Factorial and Binomial Coefficient

Definition 6 (Factorial, Dt.: Fakultät, Faktorielle)

For $n \in \mathbb{N}_0$,

$$n! := \begin{cases} 1 & \text{if } n \leq 1, \\ n \cdot (n-1)! & \text{if } n > 1. \end{cases}$$

- Note that $0! = 1$ by definition!

Definition 7 (Binomial coefficient, Dt.: Binomialkoeffizient)

Let $n \in \mathbb{N}_0$ and $k \in \mathbb{Z}$. The *binomial coefficient* $\binom{n}{k}$ of n and k is defined as follows:

$$\binom{n}{k} := \begin{cases} 0 & \text{if } k < 0, \\ \dfrac{n!}{k! \cdot (n-k)!} & \text{if } 0 \leq k \leq n, \\ 0 & \text{if } k > n. \end{cases}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Factorial and Binomial Coefficient

Definition 6 (Factorial, Dt.: Fakultät, Faktorielle)

For $n \in \mathbb{N}_0$,

$$n! := \begin{cases} 1 & \text{if } n \leq 1, \\ n \cdot (n-1)! & \text{if } n > 1. \end{cases}$$

- Note that $0! = 1$ by definition!

Definition 7 (Binomial coefficient, Dt.: Binomialkoeffizient)

Let $n \in \mathbb{N}_0$ and $k \in \mathbb{Z}$. The *binomial coefficient* $\binom{n}{k}$ of n and k is defined as follows:

$$\binom{n}{k} := \begin{cases} 0 & \text{if } k < 0, \\ \dfrac{n!}{k! \cdot (n-k)!} & \text{if } 0 \leq k \leq n, \\ 0 & \text{if } k > n. \end{cases}$$

- The binomial coefficient $\binom{n}{k}$ is pronounced as "n choose k"; Dt.: "n über k".

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Factorial and Binomial Coefficient

Lemma 8

Let $n \in \mathbb{N}_0$ and $k \in \mathbb{Z}$.

$$\binom{n}{0} = \binom{n}{n} = 1 \qquad \binom{n}{1} = \binom{n}{n-1} = n \qquad \binom{n}{k} = \binom{n}{n-k}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Factorial and Binomial Coefficient

Lemma 8

Let $n \in \mathbb{N}_0$ and $k \in \mathbb{Z}$.

$$\binom{n}{0} = \binom{n}{n} = 1 \qquad \binom{n}{1} = \binom{n}{n-1} = n \qquad \binom{n}{k} = \binom{n}{n-k}$$

Theorem 9 (Khayyam, Yang Hui, Tartaglia, Pascal)

For $n \in \mathbb{N}$ and $k \in \mathbb{Z}$,

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Theorem 10 (Binomial Theorem, Dt.: Binomischer Lehrsatz)

For all $n \in \mathbb{N}_0$ and $a, b \in \mathbb{R}$,

$$(a+b)^n = \binom{n}{0} a^n + \binom{n}{1} a^{n-1} b + \cdots + \binom{n}{n} b^n$$

or, equivalently,

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i.$$

Theorem 10 (Binomial Theorem, Dt.: Binomischer Lehrsatz)

For all $n \in \mathbb{N}_0$ and $a, b \in \mathbb{R}$,

$$(a+b)^n = \binom{n}{0} a^n + \binom{n}{1} a^{n-1} b + \cdots + \binom{n}{n} b^n$$

or, equivalently,

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i.$$

- In particular, for all $a, b \in \mathbb{R}$,
  $(a+b)^2 = a^2 + 2ab + b^2$ and $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

2 Mathematics for Geometric Modeling

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an "addition" $\oplus\colon V \times V \to V$ and a scalar "multiplication" $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an "addition" $\oplus\colon V \times V \to V$ and a scalar "multiplication" $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

1. $(V, \oplus)$ is an Abelian group.

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an "addition" $\oplus\colon V \times V \to V$ and a scalar "multiplication" $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

1. $(V, \oplus)$ is an Abelian group.
2. Distributivity: $\lambda \odot (a \oplus b) = (\lambda \odot a) \oplus (\lambda \odot b) \qquad \forall \lambda \in \mathbb{R}, \forall a, b \in V$.
3. Distributivity: $(\lambda + \mu) \odot a = (\lambda \odot a) \oplus (\mu \odot a) \qquad \forall \lambda, \mu \in \mathbb{R}, \forall a \in V$.

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an "addition" $\oplus\colon V \times V \to V$ and a scalar "multiplication" $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

1. $(V, \oplus)$ is an Abelian group.
2. Distributivity: $\lambda \odot (a \oplus b) = (\lambda \odot a) \oplus (\lambda \odot b) \qquad \forall \lambda \in \mathbb{R}, \forall a, b \in V$.
3. Distributivity: $(\lambda + \mu) \odot a = (\lambda \odot a) \oplus (\mu \odot a) \qquad \forall \lambda, \mu \in \mathbb{R}, \forall a \in V$.
4. Associativity: $\lambda \odot (\mu \odot a) = (\lambda \cdot \mu) \odot a \qquad \forall \lambda, \mu \in \mathbb{R}, \forall a \in V$.

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an "addition" $\oplus\colon V \times V \to V$ and a scalar "multiplication" $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

1. $(V, \oplus)$ is an Abelian group.
2. Distributivity: $\lambda \odot (a \oplus b) = (\lambda \odot a) \oplus (\lambda \odot b) \qquad \forall \lambda \in \mathbb{R},\ \forall a, b \in V.$
3. Distributivity: $(\lambda + \mu) \odot a = (\lambda \odot a) \oplus (\mu \odot a) \qquad \forall \lambda, \mu \in \mathbb{R},\ \forall a \in V.$
4. Associativity: $\lambda \odot (\mu \odot a) = (\lambda \cdot \mu) \odot a \qquad \forall \lambda, \mu \in \mathbb{R},\ \forall a \in V.$
5. Neutral element: $1 \odot a = a \qquad \forall a \in V.$

Vector Space

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an “addition” $\oplus\colon V \times V \to V$ and a scalar “multiplication” $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

1. $(V, \oplus)$ is an Abelian group.
2. Distributivity: $\lambda \odot (a \oplus b) = (\lambda \odot a) \oplus (\lambda \odot b) \qquad \forall \lambda \in \mathbb{R}, \forall a, b \in V$.
3. Distributivity: $(\lambda + \mu) \odot a = (\lambda \odot a) \oplus (\mu \odot a) \qquad \forall \lambda, \mu \in \mathbb{R}, \forall a \in V$.
4. Associativity: $\lambda \odot (\mu \odot a) = (\lambda \cdot \mu) \odot a \qquad \forall \lambda, \mu \in \mathbb{R}, \forall a \in V$.
5. Neutral element: $1 \odot a = a \qquad \forall a \in V$.

- In the sequel we use the same symbols $+$ and $\cdot$ for both types of operations.
- Furthermore, we postulate the standard precedence rules.

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an "addition" $\oplus\colon V \times V \to V$ and a scalar "multiplication" $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

1. $(V, \oplus)$ is an Abelian group.
2. Distributivity: $\lambda \odot (a \oplus b) = (\lambda \odot a) \oplus (\lambda \odot b) \qquad \forall \lambda \in \mathbb{R},\ \forall a, b \in V.$
3. Distributivity: $(\lambda + \mu) \odot a = (\lambda \odot a) \oplus (\mu \odot a) \qquad \forall \lambda, \mu \in \mathbb{R},\ \forall a \in V.$
4. Associativity: $\lambda \odot (\mu \odot a) = (\lambda \cdot \mu) \odot a \qquad \forall \lambda, \mu \in \mathbb{R},\ \forall a \in V.$
5. Neutral element: $1 \odot a = a \qquad \forall a \in V.$

- In the sequel we use the same symbols $+$ and $\cdot$ for both types of operations.
- Furthermore, we postulate the standard precedence rules.
- The multiplication sign is often dropped if the meaning is clear within a specific context: λa rather than $\lambda \cdot a$ or $\lambda \odot a$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 11 (Vector space, Dt.: Vektorraum)

A set V together with an "addition" $\oplus\colon V \times V \to V$ and a scalar "multiplication" $\odot\colon \mathbb{R} \times V \to V$ defines a *vector space* over $\mathbb{R}$ (with addition $+$ and multiplication $\cdot$) if the following conditions hold:

1. $(V, \oplus)$ is an Abelian group.
2. Distributivity: $\lambda \odot (a \oplus b) = (\lambda \odot a) \oplus (\lambda \odot b) \qquad \forall \lambda \in \mathbb{R}, \forall a, b \in V.$
3. Distributivity: $(\lambda + \mu) \odot a = (\lambda \odot a) \oplus (\mu \odot a) \qquad \forall \lambda, \mu \in \mathbb{R}, \forall a \in V.$
4. Associativity: $\lambda \odot (\mu \odot a) = (\lambda \cdot \mu) \odot a \qquad \forall \lambda, \mu \in \mathbb{R}, \forall a \in V.$
5. Neutral element: $1 \odot a = a \qquad \forall a \in V.$

- In the sequel we use the same symbols $+$ and $\cdot$ for both types of operations.
- Furthermore, we postulate the standard precedence rules.
- The multiplication sign is often dropped if the meaning is clear within a specific context: λa rather than $\lambda \cdot a$ or $\lambda \odot a$.
- This definition (and the subsequent ones) can be generalized by replacing $\mathbb{R}$ with an arbitrary field.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 12 (Linear combination, Dt.: Linearkombination)

Let V be a vector space over $\mathbb{R}$, and $\nu_1, \ldots, \nu_k \in V$ and $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$, for some $k \in \mathbb{N}$. The vector

$$\nu := \lambda_1 \nu_1 + \lambda_2 \nu_2 + \cdots + \lambda_k \nu_k$$

is called a *linear combination* of the vectors $\nu_1, \ldots, \nu_k$.

Definition 12 (Linear combination, Dt.: Linearkombination)

Let V be a vector space over $\mathbb{R}$, and $\nu_1, \ldots, \nu_k \in V$ and $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$, for some $k \in \mathbb{N}$. The vector

$$\nu := \lambda_1\nu_1 + \lambda_2\nu_2 + \cdots + \lambda_k\nu_k$$

is called a *linear combination* of the vectors $\nu_1, \ldots, \nu_k$.

Definition 13 (Linear hull, Dt.: lineare Hülle)

For $S \subseteq V$, with V being a vector space over $\mathbb{R}$,

$$[S] := \{\lambda_1\nu_1 + \cdots + \lambda_k\nu_k \; : \; k \in \mathbb{N}, \nu_1, \ldots, \nu_k \in S, \lambda_1, \ldots, \lambda_k \in \mathbb{R}\}$$

forms the *linear hull* of S.

Linear Combination

Definition 12 (Linear combination, Dt.: Linearkombination)

Let V be a vector space over $\mathbb{R}$, and $\nu_1, \dots, \nu_k \in V$ and $\lambda_1, \dots, \lambda_k \in \mathbb{R}$, for some $k \in \mathbb{N}$. The vector

$$\nu := \lambda_1 \nu_1 + \lambda_2 \nu_2 + \cdots + \lambda_k \nu_k$$

is called a *linear combination* of the vectors $\nu_1, \dots, \nu_k$.

Definition 13 (Linear hull, Dt.: lineare Hülle)

For $S \subseteq V$, with V being a vector space over $\mathbb{R}$,

$$[S] := \{\lambda_1 \nu_1 + \cdots + \lambda_k \nu_k \; : \; k \in \mathbb{N}, \nu_1, \dots, \nu_k \in S, \lambda_1, \dots, \lambda_k \in \mathbb{R}\}$$

forms the *linear hull* of S.

- Note: Any linear combination is formed by a finite number of vectors, even if we are allowed to pick those vectors from an infinite set!

Linear Independence

Definition 14 (Linear independence, Dt.: lineare Unabhängigkeit)

The vectors $\nu_1, \nu_2, \ldots, \nu_k$ of a vector space V over $\mathbb{R}$ are *linearly dependent* if there exist scalars $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$, not all zero, such that

$$\lambda_1\nu_1 + \lambda_2\nu_2 + \cdots + \lambda_k\nu_k = 0.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Linear Independence

Definition 14 (Linear independence, Dt.: lineare Unabhängigkeit)

The vectors $\nu_1, \nu_2, \ldots, \nu_k$ of a vector space V over $\mathbb{R}$ are *linearly dependent* if there exist scalars $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$, not all zero, such that

$$\lambda_1 \nu_1 + \lambda_2 \nu_2 + \cdots + \lambda_k \nu_k = 0.$$

Otherwise, the vectors $\nu_1, \nu_2, \ldots, \nu_k$ are *linearly independent*.

Definition 14 (Linear independence, Dt.: lineare Unabhängigkeit)

The vectors $\nu_1, \nu_2, \ldots, \nu_k$ of a vector space V over $\mathbb{R}$ are *linearly dependent* if there exist scalars $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$, not all zero, such that

$$\lambda_1 \nu_1 + \lambda_2 \nu_2 + \cdots + \lambda_k \nu_k = 0.$$

Otherwise, the vectors $\nu_1, \nu_2, \ldots, \nu_k$ are *linearly independent*.

Lemma 15

If the vectors $\nu_1, \nu_2, \ldots, \nu_k$ of a vector space V over $\mathbb{R}$ are linearly independent then

$$\lambda_1 \nu_1 + \lambda_2 \nu_2 + \cdots + \lambda_k \nu_k = 0 \quad \Rightarrow \quad \lambda_1 = \lambda_2 = \cdots = \lambda_k = 0$$

for all $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Linear Independence

Definition 14 (Linear independence, Dt.: lineare Unabhängigkeit)

The vectors $\nu_1, \nu_2, \ldots, \nu_k$ of a vector space V over $\mathbb{R}$ are *linearly dependent* if there exist scalars $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$, not all zero, such that

$$\lambda_1\nu_1 + \lambda_2\nu_2 + \cdots + \lambda_k\nu_k = 0.$$

Otherwise, the vectors $\nu_1, \nu_2, \ldots, \nu_k$ are *linearly independent*.

Lemma 15

If the vectors $\nu_1, \nu_2, \ldots, \nu_k$ of a vector space V over $\mathbb{R}$ are linearly independent then

$$\lambda_1\nu_1 + \lambda_2\nu_2 + \cdots + \lambda_k\nu_k = 0 \quad \Rightarrow \quad \lambda_1 = \lambda_2 = \cdots = \lambda_k = 0$$

for all $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$.

Lemma 16

The vectors $\nu_1, \nu_2, \ldots, \nu_k$ of a vector space V over $\mathbb{R}$ are linearly independent if and only if none of them can be expressed as a linear combination of the other vectors.

Basis of a Vector Space

Definition 17 (Basis)

The vectors $\nu_1, \nu_2, \ldots, \nu_n \in V$ form a *basis* of the vector space V over $\mathbb{R}$ if

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Basis of a Vector Space

Definition 17 (Basis)

The vectors $\nu_1, \nu_2, \ldots, \nu_n \in V$ form a *basis* of the vector space V over $\mathbb{R}$ if

1. $\nu_1, \ldots, \nu_n$ are linearly independent;
2. $[\{\nu_1, \ldots, \nu_n\}] = V$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Basis of a Vector Space

Definition 17 (Basis)

The vectors $\nu_1, \nu_2, \ldots, \nu_n \in V$ form a *basis* of the vector space V over $\mathbb{R}$ if

1. $\nu_1, \ldots, \nu_n$ are linearly independent;
2. $[\{\nu_1, \ldots, \nu_n\}] = V$.

Definition 18 (Finite dimension)

A vector space V is said to have *finite dimension* if their exists a basis of V that has finitely many vectors.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Basis of a Vector Space

Definition 17 (Basis)

The vectors $\nu_1, \nu_2, \ldots, \nu_n \in V$ form a *basis* of the vector space V over $\mathbb{R}$ if

1. $\nu_1, \ldots, \nu_n$ are linearly independent;
2. $[\{\nu_1, \ldots, \nu_n\}] = V$.

Definition 18 (Finite dimension)

A vector space V is said to have *finite dimension* if their exists a basis of V that has finitely many vectors.

Theorem 19

Every basis of a finite vector space has the same number of basis vectors.

Basis of a Vector Space

Definition 17 (Basis)

The vectors $\nu_1, \nu_2, \ldots, \nu_n \in V$ form a *basis* of the vector space V over $\mathbb{R}$ if

1. $\nu_1, \ldots, \nu_n$ are linearly independent;
2. $[\{\nu_1, \ldots, \nu_n\}] = V$.

Definition 18 (Finite dimension)

A vector space V is said to have *finite dimension* if their exists a basis of V that has finitely many vectors.

Theorem 19

Every basis of a finite vector space has the same number of basis vectors.

- The number of vectors of a basis is called the *dimension* of the vector space.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Basis of a Vector Space

Definition 17 (Basis)

The vectors $\nu_1, \nu_2, \ldots, \nu_n \in V$ form a *basis* of the vector space V over $\mathbb{R}$ if

1. $\nu_1, \ldots, \nu_n$ are linearly independent;
2. $[\{\nu_1, \ldots, \nu_n\}] = V$.

Definition 18 (Finite dimension)

A vector space V is said to have *finite dimension* if their exists a basis of V that has finitely many vectors.

Theorem 19

Every basis of a finite vector space has the same number of basis vectors.

- The number of vectors of a basis is called the *dimension* of the vector space.

Theorem 20

If $\nu_1, \ldots, \nu_n$ form a basis for V over $\mathbb{R}$ then for all $\nu \in V$ exist uniquely determined $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ such that $\nu = \lambda_1\nu_1 + \lambda_2\nu_2 + \cdots + \lambda_n\nu_n$.

Mathematics for Geometric Modeling

Convex Combination

Definition 21 (Convex combination, Dt.: Konvexkombination)

Let $p_1, p_2, \ldots, p_k$ be k points in $\mathbb{R}^n$. A *convex combination* of $p_1, \ldots, p_k$ is given by

$$\sum_{i=1}^{k} \lambda_i p_i \quad \text{with} \quad \sum_{i=1}^{k} \lambda_i = 1 \quad \text{and} \quad \forall (1 \leq i \leq k) \;\; \lambda_i \geq 0,$$

where $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}$ are scalars.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Convex Combination

Definition 21 (Convex combination, Dt.: Konvexkombination)

Let $p_1, p_2, \ldots, p_k$ be k points in $\mathbb{R}^n$. A *convex combination* of $p_1, \ldots, p_k$ is given by

$$\sum_{i=1}^{k} \lambda_i p_i \quad \text{with} \quad \sum_{i=1}^{k} \lambda_i = 1 \quad \text{and} \quad \forall(1 \leq i \leq k) \;\; \lambda_i \geq 0,$$

where $\lambda_1, \lambda_2, \ldots, \lambda_k \in \mathbb{R}$ are scalars.

- Hence, a convex combination is a linear combination (of the position vectors) of the points with the added restrictions

$$\forall(1 \leq i \leq k) \;\; \lambda_i \geq 0 \qquad \text{and} \qquad \sum_{i=1}^{k} \lambda_i = 1.$$

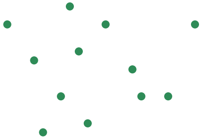
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Convex Hull

Definition 22 (Convex hull, Dt.: konvexe Hülle)

Let $p_1, p_2, \ldots, p_k$ be k points in $\mathbb{R}^n$. The *convex hull* of $p_1, \ldots, p_k$ is the set

$$\{\sum_{i=1}^{k} \lambda_i p_i : \ \lambda_1, \ldots \lambda_k \in \mathbb{R}_0^+ \text{ and } \sum_{i=1}^{k} \lambda_i \ = \ 1\}.$$

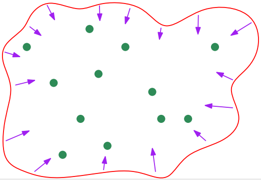


Convex Hull

Definition 22 (Convex hull, Dt.: konvexe Hülle)

Let $p_1, p_2, \ldots, p_k$ be k points in $\mathbb{R}^n$. The *convex hull* of $p_1, \ldots, p_k$ is the set

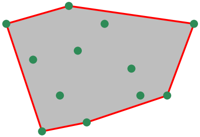
$$\{\sum_{i=1}^{k} \lambda_i p_i : \lambda_1, \ldots \lambda_k \in \mathbb{R}_0^+ \text{ and } \sum_{i=1}^{k} \lambda_i = 1\}.$$



Definition 22 (Convex hull, Dt.: konvexe Hülle)

Let $p_1, p_2, \ldots, p_k$ be k points in $\mathbb{R}^n$. The *convex hull* of $p_1, \ldots, p_k$ is the set

$$\{\sum_{i=1}^{k} \lambda_i p_i : \lambda_1, \ldots \lambda_k \in \mathbb{R}_0^+ \text{ and } \sum_{i=1}^{k} \lambda_i = 1\}.$$



Definition 22 (Convex hull, Dt.: konvexe Hülle)

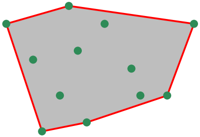
Let $p_1, p_2, \ldots, p_k$ be k points in $\mathbb{R}^n$. The *convex hull* of $p_1, \ldots, p_k$ is the set

$$\{\sum_{i=1}^{k} \lambda_i p_i : \lambda_1, \ldots \lambda_k \in \mathbb{R}_0^+ \text{ and } \sum_{i=1}^{k} \lambda_i = 1\}.$$

For a set $S \subseteq \mathbb{R}^n$ (with possibly infinitely many points), the *convex hull* of S is the set

$$\{\sum_{i=1}^{k} \lambda_i p_i : k \in \mathbb{N} \text{ and } p_1, p_2, \ldots, p_k \in S \text{ and } \lambda_1, \ldots \lambda_k \in \mathbb{R}_0^+ \text{ and } \sum_{i=1}^{k} \lambda_i = 1\}.$$

The convex hull of S is commonly denoted by $CH(S)$.



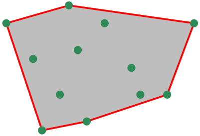
Convexity

Definition 23 (Convex set, Dt.: konvexe Menge)

A set $S \subseteq \mathbb{R}^n$ is called *convex* if for all $p, q \in S$

$$\overline{pq} \subseteq S,$$

where $\overline{pq}$ denotes the straight-line segment between p and q.



Definition 23 (Convex set, Dt.: konvexe Menge)

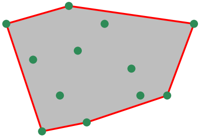
A set $S \subseteq \mathbb{R}^n$ is called *convex* if for all $p, q \in S$

$$\overline{pq} \subseteq S,$$

where $\overline{pq}$ denotes the straight-line segment between p and q.

Lemma 24

For $S \subseteq \mathbb{R}^n$, the convex hull $CH(S)$ of S is a convex set.



Definition 23 (Convex set, Dt.: konvexe Menge)

A set $S \subseteq \mathbb{R}^n$ is called *convex* if for all $p, q \in S$

$$\overline{pq} \subseteq S,$$

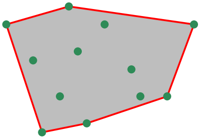
where $\overline{pq}$ denotes the straight-line segment between p and q.

Lemma 24

For $S \subseteq \mathbb{R}^n$, the convex hull $CH(S)$ of S is a convex set.

Lemma 25

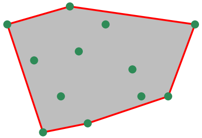
For a set S of n points in $\mathbb{R}^2$, the convex hull $CH(S)$ is a convex polygon.



Definition 26 (Convex superset)

A set $B \subseteq \mathbb{R}^n$ is called a *convex superset* of a set $A \subseteq \mathbb{R}^n$ if

$$A \subseteq B \quad \text{and} \quad B \text{ is convex.}$$



Definition 26 (Convex superset)

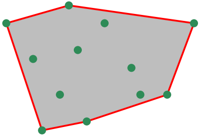
A set $B \subseteq \mathbb{R}^n$ is called a *convex superset* of a set $A \subseteq \mathbb{R}^n$ if

$$A \subseteq B \quad \text{and} \quad B \text{ is convex.}$$

Lemma 27

For $A \subseteq \mathbb{R}^n$, the following definitions are equivalent to Def. 22:

- $CH(A)$ is the smallest convex superset of A.
- $CH(A)$ is the intersection of all convex supersets of A.



Definition 26 (Convex superset)

A set $B \subseteq \mathbb{R}^n$ is called a *convex superset* of a set $A \subseteq \mathbb{R}^n$ if

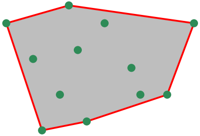
$$A \subseteq B \quad \text{and} \quad B \text{ is convex.}$$

Lemma 27

For $A \subseteq \mathbb{R}^n$, the following definitions are equivalent to Def. 22:

- $CH(A)$ is the smallest convex superset of A.
- $CH(A)$ is the intersection of all convex supersets of A.

- The definition of a convex hull (and of convexity) is readily extended from $\mathbb{R}^n$ to other vector spaces over $\mathbb{R}$.



2 Mathematics for Geometric Modeling

Definition 28 (Monomial, Dt.: Monom)

For $m \in \mathbb{N}$, a (real) *monomial* in m variables $x_1, x_2, \ldots, x_m$ is a product of a coefficient $c \in \mathbb{R}$ and powers of the variables x_i with exponents $k_i \in \mathbb{N}_0$:

$$c \prod_{i=1}^{m} x_i^{k_i} = c \cdot x_1^{k_1} \cdot x_2^{k_2} \cdot \ldots \cdot x_m^{k_m}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 28 (Monomial, Dt.: Monom)

For $m \in \mathbb{N}$, a (real) *monomial* in m variables $x_1, x_2, \ldots, x_m$ is a product of a coefficient $c \in \mathbb{R}$ and powers of the variables x_i with exponents $k_i \in \mathbb{N}_0$:

$$c \prod_{i=1}^{m} x_i^{k_i} = c \cdot x_1^{k_1} \cdot x_2^{k_2} \cdot \ldots \cdot x_m^{k_m}.$$

The *degree of the monomial* is given by $\sum_{i=1}^{m} k_i$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomials

Definition 28 (Monomial, Dt.: Monom)

For $m \in \mathbb{N}$, a (real) *monomial* in m variables $x_1, x_2, \ldots, x_m$ is a product of a coefficient $c \in \mathbb{R}$ and powers of the variables x_i with exponents $k_i \in \mathbb{N}_0$:

$$c \prod_{i=1}^{m} x_i^{k_i} = c \cdot x_1^{k_1} \cdot x_2^{k_2} \cdot \ldots \cdot x_m^{k_m}.$$

The *degree of the monomial* is given by $\sum_{i=1}^{m} k_i$.

Definition 29 (Polynomial, Dt.: Polynom)

For $m \in \mathbb{N}$, a (real) *polynomial* in m variables $x_1, x_2, \ldots, x_m$ is a finite sum of monomials in $x_1, x_2, \ldots, x_m$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomials

Definition 28 (Monomial, Dt.: Monom)

For $m \in \mathbb{N}$, a (real) *monomial* in m variables $x_1, x_2, \ldots, x_m$ is a product of a coefficient $c \in \mathbb{R}$ and powers of the variables x_i with exponents $k_i \in \mathbb{N}_0$:

$$c \prod_{i=1}^{m} x_i^{k_i} = c \cdot x_1^{k_1} \cdot x_2^{k_2} \cdot \ldots \cdot x_m^{k_m}.$$

The *degree of the monomial* is given by $\sum_{i=1}^{m} k_i$.

Definition 29 (Polynomial, Dt.: Polynom)

For $m \in \mathbb{N}$, a (real) *polynomial* in m variables $x_1, x_2, \ldots, x_m$ is a finite sum of monomials in $x_1, x_2, \ldots, x_m$.
A polynomial is *univariate* if $m = 1$, *bivariate* if $m = 2$, and *multivariate* otherwise.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomials

Definition 28 (Monomial, Dt.: Monom)

For $m \in \mathbb{N}$, a (real) *monomial* in m variables $x_1, x_2, \ldots, x_m$ is a product of a coefficient $c \in \mathbb{R}$ and powers of the variables x_i with exponents $k_i \in \mathbb{N}_0$:

$$c \prod_{i=1}^{m} x_i^{k_i} = c \cdot x_1^{k_1} \cdot x_2^{k_2} \cdot \ldots \cdot x_m^{k_m}.$$

The *degree of the monomial* is given by $\sum_{i=1}^{m} k_i$.

Definition 29 (Polynomial, Dt.: Polynom)

For $m \in \mathbb{N}$, a (real) *polynomial* in m variables $x_1, x_2, \ldots, x_m$ is a finite sum of monomials in $x_1, x_2, \ldots, x_m$.
A polynomial is *univariate* if $m = 1$, *bivariate* if $m = 2$, and *multivariate* otherwise.

Definition 30 (Degree, Dt.: Grad)

The *degree of a polynomial* is the maximum degree of its monomials.

Polynomials

- Hence, a univariate polynomial over $\mathbb{R}$ with variable x of degree n is a term of the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

  with coefficients $a_0, \ldots, a_n \in \mathbb{R}$ and $a_n \neq 0$.
- It is a convention to drop all monomials whose coefficients are zero.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomials

- Hence, a univariate polynomial over $\mathbb{R}$ with variable x of degree n is a term of the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

with coefficients $a_0, \ldots, a_n \in \mathbb{R}$ and $a_n \neq 0$.

- It is a convention to drop all monomials whose coefficients are zero.
- Univariate polynomials are usually written according to a decreasing order of the exponents of their monomials.
- In that case, the first term is the *leading term* which indicates the degree of the polynomial; its coefficient is the *leading coefficient*.

Polynomials

- Hence, a univariate polynomial over $\mathbb{R}$ with variable x of degree n is a term of the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

with coefficients $a_0, \ldots, a_n \in \mathbb{R}$ and $a_n \neq 0$.
- It is a convention to drop all monomials whose coefficients are zero.
- Univariate polynomials are usually written according to a decreasing order of the exponents of their monomials.
- In that case, the first term is the *leading term* which indicates the degree of the polynomial; its coefficient is the *leading coefficient*.
- Univariate polynomials of degree
  0. are called constant polynomials,
  1. are called linear polynomials,
  2. are called quadratic polynomials,
  3. are called cubic polynomials,
  4. are called quartic polynomials,
  5. are called quintic polynomials.

Polynomials

- Hence, a univariate polynomial over $\mathbb{R}$ with variable x of degree n is a term of the form

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0,$$

with coefficients $a_0, \ldots, a_n \in \mathbb{R}$ and $a_n \neq 0$.
- It is a convention to drop all monomials whose coefficients are zero.
- Univariate polynomials are usually written according to a decreasing order of the exponents of their monomials.
- In that case, the first term is the *leading term* which indicates the degree of the polynomial; its coefficient is the *leading coefficient*.
- Univariate polynomials of degree
  0. are called constant polynomials,
  1. are called linear polynomials,
  2. are called quadratic polynomials,
  3. are called cubic polynomials,
  4. are called quartic polynomials,
  5. are called quintic polynomials.
- The set of all univariate polynomials with variable x and coefficients out of $\mathbb{R}$ is denoted by $\mathbb{R}[x]$. Similarly, $\mathbb{R}[x, y]$ for all bivariate polynomials in x and y.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomial Arithmetic

- We define the addition of (univariate) polynomials based on the pairwise addition of corresponding coefficients:

$$\left(\sum_{i=0}^{n} a_i x^i\right) + \left(\sum_{i=0}^{n} b_i x^i\right) := \sum_{i=0}^{n} (a_i + b_i) x^i$$

Polynomial Arithmetic

- We define the addition of (univariate) polynomials based on the pairwise addition of corresponding coefficients:

$$\left(\sum_{i=0}^{n} a_i x^i\right) + \left(\sum_{i=0}^{n} b_i x^i\right) := \sum_{i=0}^{n} (a_i + b_i) x^i$$

- The multiplication of polynomials is based on the multiplication within $\mathbb{R}$, distributivity, and the rules

$$ax = xa \quad \text{and} \quad x^m x^k = x^{m+k}$$

for all $a \in \mathbb{R}$ and $m, k \in \mathbb{N}$:

$$\left(\sum_{i=0}^{n} a_i x^i\right) \cdot \left(\sum_{j=0}^{m} b_j x^j\right) := \sum_{i=0}^{n} \sum_{j=0}^{m} (a_i b_j) x^{i+j}$$

Polynomial Arithmetic

- We define the addition of (univariate) polynomials based on the pairwise addition of corresponding coefficients:

$$\left(\sum_{i=0}^{n} a_i x^i\right) + \left(\sum_{i=0}^{n} b_i x^i\right) := \sum_{i=0}^{n} (a_i + b_i) x^i$$

- The multiplication of polynomials is based on the multiplication within $\mathbb{R}$, distributivity, and the rules

$$ax = xa \quad \text{and} \quad x^m x^k = x^{m+k}$$

for all $a \in \mathbb{R}$ and $m, k \in \mathbb{N}$:

$$\left(\sum_{i=0}^{n} a_i x^i\right) \cdot \left(\sum_{j=0}^{m} b_j x^j\right) := \sum_{i=0}^{n} \sum_{j=0}^{m} (a_i b_j) x^{i+j}$$

- Elementary properties of polynomials: One can prove easily that the addition, multiplication and composition of two polynomials as well as their derivative and antiderivative (indefinite integral) again yield a polynomial.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomial Arithmetic

- We define the addition of (univariate) polynomials based on the pairwise addition of corresponding coefficients:

$$\left(\sum_{i=0}^{n} a_i x^i\right) + \left(\sum_{i=0}^{n} b_i x^i\right) := \sum_{i=0}^{n} (a_i + b_i) x^i$$

- The multiplication of polynomials is based on the multiplication within $\mathbb{R}$, distributivity, and the rules

$$ax = xa \quad \text{and} \quad x^m x^k = x^{m+k}$$

for all $a \in \mathbb{R}$ and $m, k \in \mathbb{N}$:

$$\left(\sum_{i=0}^{n} a_i x^i\right) \cdot \left(\sum_{j=0}^{m} b_j x^j\right) := \sum_{i=0}^{n} \sum_{j=0}^{m} (a_i b_j) x^{i+j}$$

- Elementary properties of polynomials: One can prove easily that the addition, multiplication and composition of two polynomials as well as their derivative and antiderivative (indefinite integral) again yield a polynomial.
- Similarly for multivariate polynomials.

Theorem 31

The univariate polynomials of $\mathbb{R}[x]$ form an infinite vector space over $\mathbb{R}$. The so-called *power basis* of this vector space is given by the monomials $1, x, x^2, x^3, \ldots$.

Polynomials: Vector Space

Theorem 31

The univariate polynomials of $\mathbb{R}[x]$ form an infinite vector space over $\mathbb{R}$. The so-called *power basis* of this vector space is given by the monomials $1, x, x^2, x^3, \ldots$.

Lemma 32

The monomials $1, x, x^2, x^3, \ldots, x^n$ form a basis of the vector space of polynomials of degree up to n over $\mathbb{R}$, for all $n \in \mathbb{N}_0$.

Polynomials: Vector Space

Theorem 31

The univariate polynomials of $\mathbb{R}[x]$ form an infinite vector space over $\mathbb{R}$. The so-called *power basis* of this vector space is given by the monomials $1, x, x^2, x^3, \ldots$.

Lemma 32

The monomials $1, x, x^2, x^3, \ldots, x^n$ form a basis of the vector space of polynomials of degree up to n over $\mathbb{R}$, for all $n \in \mathbb{N}_0$.

- The power basis is not the only meaningful basis of the polynomials of $\mathbb{R}[x]$. See, e.g., the Bernstein polynomials that are used to form Bézier curves.

Polynomial: Evaluation

- Consider a polynomial $p \in \mathbb{R}[x]$ of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

Polynomial: Evaluation

- Consider a polynomial $p \in \mathbb{R}[x]$ of degree n with coefficients $a_0, a_1, \dots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

- A straightforward polynomial evaluation of p for a given parameter x_0 results in k multiplications for a monomial of degree k, plus a total of n additions.
- Hence, we would get

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$$

multiplications (and n additions).

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomial: Evaluation

- Consider a polynomial $p \in \mathbb{R}[x]$ of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

- A straightforward polynomial evaluation of p for a given parameter x_0 results in k multiplications for a monomial of degree k, plus a total of n additions.
- Hence, we would get

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$$

multiplications (and n additions).
- Can we do better?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Polynomial: Evaluation

- Consider a polynomial $p \in \mathbb{R}[x]$ of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

- A straightforward polynomial evaluation of p for a given parameter x_0 results in k multiplications for a monomial of degree k, plus a total of n additions.
- Hence, we would get

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$$

  multiplications (and n additions).
- Can we do better?
- Obviously, we can reduce the number of multiplications to $O(n \log n)$ by resorting to exponentiation by squaring:

$$x^n := \begin{cases} x\,(x^2)^{\frac{n-1}{2}} & \text{if } n \text{ is odd,} \\ (x^2)^{\frac{n}{2}} & \text{if } n \text{ is even.} \end{cases}$$

Polynomial: Evaluation

- Consider a polynomial $p \in \mathbb{R}[x]$ of degree n with coefficients $a_0, a_1, \ldots, a_n \in \mathbb{R}$, with $a_n \neq 0$:

$$p(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n.$$

- A straightforward polynomial evaluation of p for a given parameter x_0 results in k multiplications for a monomial of degree k, plus a total of n additions.
- Hence, we would get

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$$

multiplications (and n additions).

- Can we do better?
- Obviously, we can reduce the number of multiplications to $O(n \log n)$ by resorting to exponentiation by squaring:

$$x^n := \begin{cases} x\,(x^2)^{\frac{n-1}{2}} & \text{if } n \text{ is odd,} \\ (x^2)^{\frac{n}{2}} & \text{if } n \text{ is even.} \end{cases}$$

- Can we do even better?

Polynomial: Horner's Algorithm

- *Horner's Algorithm*: The idea is to rewrite the polynomial such that

$$p(x) = a_0 + x\Big(a_1 + x(a_2 + \ldots + x(a_{n-2} + x(a_{n-1} + x\, a_n)) \ldots)\Big)$$

Polynomial: Horner's Algorithm

- *Horner's Algorithm*: The idea is to rewrite the polynomial such that

$$p(x) = a_0 + x\Big(a_1 + x(a_2 + \ldots + x(a_{n-2} + x(a_{n-1} + x\, a_n))\ldots)\Big)$$

and compute the result $h_0 := p(x_0)$ as follows:

$$h_n := a_n$$

$$h_i := x_0 \cdot h_{i+1} + a_i \qquad \text{for } i := n-1, n-2, \ldots, 2, 1, 0$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- *Horner's Algorithm*: The idea is to rewrite the polynomial such that

$$p(x) = a_0 + x\Big(a_1 + x\big(a_2 + \ldots + x(a_{n-2} + x(a_{n-1} + x\,a_n))\ldots\big)\Big)$$

  and compute the result $h_0 := p(x_0)$ as follows:

$$h_n := a_n$$
$$h_i := x_0 \cdot h_{i+1} + a_i \qquad \text{for } i := n-1, n-2, \ldots, 2, 1, 0$$

```c
/** Evaluates a polynomial of degree n at point x
 * @param p: array of n+1 coefficients
 * @param n: the degree of the polynomial
 * @param x: the point of evaluation
 * @return the evaluation result
 */
double evaluate(double *p, int n, double x)
{
  double h = p[n];

  for (int i = n - 1; i >= 0; --i)
    h = x * h + p[i];

  return h;
}
```

Lemma 33

Horner's Algorithm consumes n multiplications and n additions to evaluate a polynomial of degree n.

Lemma 33

Horner's Algorithm consumes n multiplications and n additions to evaluate a polynomial of degree n.

Caveat

Subtractive cancellation could occur at any time, and there is no easy way to determine a priori whether and for which data it will indeed occur.

- Subtractive cancellation: Subtracting two nearly equal numbers (on a conventional IEEE-754 floating-point arithmetic) may yield a result with few or no meaningful digits. Aka: catastrophic cancellation.

2 Mathematics for Geometric Modeling

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 34 (Derivative, Dt.: Ableitung)

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable* at an interior point $x_0 \in S$ if

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

exists, in which case the limit is called the *derivative* of f at x_0, denoted by $f'(x_0)$.

Differentiation of Functions of One Variable

Definition 34 (Derivative, Dt.: Ableitung)

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable* at an interior point $x_0 \in S$ if

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

exists, in which case the limit is called the *derivative* of f at x_0, denoted by $f'(x_0)$.

Definition 35

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable on S* if it is differentiable at every point of S.

Definition 34 (Derivative, Dt.: Ableitung)

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable* at an interior point $x_0 \in S$ if

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

exists, in which case the limit is called the *derivative* of f at x_0, denoted by $f'(x_0)$.

Definition 35

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable on S* if it is differentiable at every point of S.
If f is differentiable on S and f' is continuous on S then f is *continuously differentiable on S*. In this case f is said to be of *differentiability class C^1*.

Definition 34 (Derivative, Dt.: Ableitung)

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable* at an interior point $x_0 \in S$ if

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

exists, in which case the limit is called the *derivative* of f at x_0, denoted by $f'(x_0)$.

Definition 35

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable on S* if it is differentiable at every point of S.
If f is differentiable on S and f' is continuous on S then f is *continuously differentiable on S*. In this case f is said to be of *differentiability class C^1*.

- By taking one-sided limits one can also consider one-sided derivatives on the boundary of closed sets S.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 34 (Derivative, Dt.: Ableitung)

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable* at an interior point $x_0 \in S$ if

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

exists, in which case the limit is called the *derivative* of f at x_0, denoted by $f'(x_0)$.

Definition 35

Let $S \subseteq \mathbb{R}$ be an open set. A (scalar-valued) function $f\colon S \to \mathbb{R}$ is *differentiable on S* if it is differentiable at every point of S.
If f is differentiable on S and f' is continuous on S then f is *continuously differentiable on S*. In this case f is said to be of *differentiability class C^1*.

- By taking one-sided limits one can also consider one-sided derivatives on the boundary of closed sets S.
- By applying differentiation to f', a second derivative f'' of f can be defined. Inductively, we obtain $f^{(n)}$ by differentiating $f^{(n-1)}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 36 (C^k, Dt.: k-mal stetig differenzierbar)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ that has k successive derivatives is called k *times differentiable*. If, in addition, the k-th derivative is continuous, then the function is said to be of *differentiability class* C^k.

Definition 36 (C^k, Dt.: k-mal stetig differenzierbar)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ that has k successive derivatives is called k *times differentiable*. If, in addition, the k-th derivative is continuous, then the function is said to be of *differentiability class* C^k.

- If the k-th derivative of f exists then the continuity of $f^{(0)}, f^{(1)}, \ldots, f^{(k-1)}$ is implied.

Definition 36 (C^k, Dt.: k-mal stetig differenzierbar)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ that has k successive derivatives is called k *times differentiable*. If, in addition, the k-th derivative is continuous, then the function is said to be of *differentiability class* C^k.

- If the k-th derivative of f exists then the continuity of $f^{(0)}, f^{(1)}, \ldots, f^{(k-1)}$ is implied.

Definition 37 (Smooth, Dt.: glatt)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ is called *smooth* if it has infinitely many derivatives, denoted by the class C^∞.

Definition 36 (C^k, Dt.: k-mal stetig differenzierbar)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ that has k successive derivatives is called k *times differentiable*. If, in addition, the k-th derivative is continuous, then the function is said to be of *differentiability class* C^k.

- If the k-th derivative of f exists then the continuity of $f^{(0)}, f^{(1)}, \ldots, f^{(k-1)}$ is implied.

Definition 37 (Smooth, Dt.: glatt)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ is called *smooth* if it has infinitely many derivatives, denoted by the class C^∞.

- We have $C^\infty \subset C^i \subset C^j$, for all $i, j \in \mathbb{N}_0$ if $i > j$.

Definition 36 (C^k, Dt.: *k*-mal stetig differenzierbar)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ that has k successive derivatives is called *k times differentiable*. If, in addition, the k-th derivative is continuous, then the function is said to be of *differentiability class C^k*.

- If the k-th derivative of f exists then the continuity of $f^{(0)}, f^{(1)}, \dots, f^{(k-1)}$ is implied.

Definition 37 (Smooth, Dt.: glatt)

Let $S \subseteq \mathbb{R}$ be an open set. A function $f\colon S \to \mathbb{R}$ is called *smooth* if it has infinitely many derivatives, denoted by the class C^∞.

- We have $C^\infty \subset C^i \subset C^j$, for all $i, j \in \mathbb{N}_0$ if $i > j$.
- Notation:
  - $f^{(0)}(x) := f(x)$ for convenience purposes.
  - $f'(x) = f^{(1)}(x) = \frac{d}{dx} f(x) = \frac{df}{dx}(x)$.
  - $f''(x) = f^{(2)}(x) = \frac{d^2}{dx^2} f(x) = \frac{d^2 f}{dx^2}(x)$.
  - $f'''(x) = f^{(3)}(x) = \frac{d^3}{dx^3} f(x) = \frac{d^3 f}{dx^3}(x)$.
  - $f^{(n)}(x) = \frac{d^n}{dx^n} f(x) = \frac{d^n f}{dx^n}(x)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 38

For $n \in \mathbb{N}$ consider n functions $f_i\colon S \to \mathbb{R}$ (with $1 \le i \le n$) and define $f\colon S \to \mathbb{R}^n$ as

$$f(x) := \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 38

For $n \in \mathbb{N}$ consider n functions $f_i\colon S \to \mathbb{R}$ (with $1 \le i \le n$) and define $f\colon S \to \mathbb{R}^n$ as

$$f(x) := \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}.$$

Then the (vector-valued) function f is *differentiable* at an interior point $x_0 \in S$ if and only if f_i is differentiable at x_0, for all $i \in \{1, 2, \ldots, n\}$.

Definition 38

For $n \in \mathbb{N}$ consider n functions $f_i\colon S \to \mathbb{R}$ (with $1 \le i \le n$) and define $f\colon S \to \mathbb{R}^n$ as

$$f(x) := \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}.$$

Then the (vector-valued) function f is *differentiable* at an interior point $x_0 \in S$ if and only if f_i is differentiable at x_0, for all $i \in \{1, 2, \ldots, n\}$. The derivative of f at x_0 is given by

$$f'(x_0) := \begin{pmatrix} f_1'(x_0) \\ f_2'(x_0) \\ \vdots \\ f_n'(x_0) \end{pmatrix}.$$

Definition 38

For $n \in \mathbb{N}$ consider n functions $f_i\colon S \to \mathbb{R}$ (with $1 \le i \le n$) and define $f\colon S \to \mathbb{R}^n$ as

$$f(x) := \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}.$$

Then the (vector-valued) function f is *differentiable* at an interior point $x_0 \in S$ if and only if f_i is differentiable at x_0, for all $i \in \{1, 2, \ldots, n\}$. The derivative of f at x_0 is given by

$$f'(x_0) := \begin{pmatrix} f_1'(x_0) \\ f_2'(x_0) \\ \vdots \\ f_n'(x_0) \end{pmatrix}.$$

- All other definitions related to differentiability carry over from scalar-valued functions to vector-valued functions of one variable in a natural way.

Definition 39 (Partial derivative, Dt.: partielle Ableitung)

Let $S \subseteq \mathbb{R}^m$ be an open set. The *partial derivative* of a (vector-valued) function $f\colon S \to \mathbb{R}^n$ at point $(a_1, a_2, \dots, a_m) \in S$ with respect to the i-th coordinate x_i is defined as

$$\frac{\partial f}{\partial x_i}(a_1, a_2, \dots, a_m) := \lim_{h \to 0} \frac{f(a_1, a_2, \dots, a_i + h, \dots, a_m) - f(a_1, a_2, \dots, a_i, \dots, a_m)}{h},$$

if this limit exists.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 39 (Partial derivative, Dt.: partielle Ableitung)

Let $S \subseteq \mathbb{R}^m$ be an open set. The *partial derivative* of a (vector-valued) function $f\colon S \to \mathbb{R}^n$ at point $(a_1, a_2, \ldots, a_m) \in S$ with respect to the i-th coordinate x_i is defined as

$$\frac{\partial f}{\partial x_i}(a_1, a_2, \ldots, a_m) := \lim_{h \to 0} \frac{f(a_1, a_2, \ldots, a_i + h, \ldots, a_m) - f(a_1, a_2, \ldots, a_i, \ldots, a_m)}{h},$$

if this limit exists.

- Hence, for a partial derivative with respect to x_i we simply differentiate f with respect to x_i according to the rules for ordinary differentiation, while regarding all other variables as constants.
- That is, for the purpose of the partial derivative with respect to x_i we regard f as univariate function in x_i and apply standard differentiation rules.

Definition 39 (Partial derivative, Dt.: partielle Ableitung)

Let $S \subseteq \mathbb{R}^m$ be an open set. The *partial derivative* of a (vector-valued) function $f\colon S \to \mathbb{R}^n$ at point $(a_1, a_2, \ldots, a_m) \in S$ with respect to the i-th coordinate x_i is defined as

$$\frac{\partial f}{\partial x_i}(a_1, a_2, \ldots, a_m) := \lim_{h \to 0} \frac{f(a_1, a_2, \ldots, a_i + h, \ldots, a_m) - f(a_1, a_2, \ldots, a_i, \ldots, a_m)}{h},$$

if this limit exists.

- Hence, for a partial derivative with respect to x_i we simply differentiate f with respect to x_i according to the rules for ordinary differentiation, while regarding all other variables as constants.
- That is, for the purpose of the partial derivative with respect to x_i we regard f as univariate function in x_i and apply standard differentiation rules.
- Some authors prefer to write f_x instead of $\frac{\partial f}{\partial x}$.
- We will mix notations as we find it convenient.

Note

A function of m variables may have all first-order partial derivatives at a point $(a_1, \ldots, a_m)$ but still need not be continuous at $(a_1, \ldots, a_m)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Note

A function of m variables may have all first-order partial derivatives at a point $(a_1, \ldots, a_m)$ but still need not be continuous at $(a_1, \ldots, a_m)$.

Definition 40 (Continuously differentiable, Dt.: stetig differenzierbar)

We say that a function $f\colon S \to \mathbb{R}^n$ of m variables is *continuously differentiable* on an open subset S of $\mathbb{R}^m$ if $\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_m}$ exist and are continuous on S.

2 Mathematics for Geometric Modeling

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Curves

- Intuitively, a curve in $\mathbb{R}^2$ is generated by a continuous motion of a pencil on a sheet of paper.
- A formal mathematical definition is not entirely straightforward, and the term "curve" is associated with two closely related notions: kinematic and geometric.

- Intuitively, a curve in $\mathbb{R}^2$ is generated by a continuous motion of a pencil on a sheet of paper.
- A formal mathematical definition is not entirely straightforward, and the term “curve” is associated with two closely related notions: kinematic and geometric.
- In the kinematic setting, a (parametric) curve is a function of one real variable.

Curves

- Intuitively, a curve in $\mathbb{R}^2$ is generated by a continuous motion of a pencil on a sheet of paper.
- A formal mathematical definition is not entirely straightforward, and the term "curve" is associated with two closely related notions: kinematic and geometric.
- In the kinematic setting, a (parametric) curve is a function of one real variable.
- In the geometric setting, a curve, also called an arc, is a 1-dimensional subset of space that is "similar" to a line (albeit it need not be straight).
- E.g., the unit circle in the (Euclidean) plane can be defined algebraically as the zero set of the equation $x^2 + y^2 - 1 = 0$, for $x, y \in \mathbb{R}$.

Curves

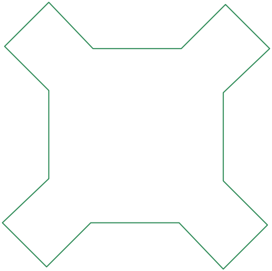
- Intuitively, a curve in $\mathbb{R}^2$ is generated by a continuous motion of a pencil on a sheet of paper.
- A formal mathematical definition is not entirely straightforward, and the term "curve" is associated with two closely related notions: kinematic and geometric.
- In the kinematic setting, a (parametric) curve is a function of one real variable.
- In the geometric setting, a curve, also called an arc, is a 1-dimensional subset of space that is "similar" to a line (albeit it need not be straight).
- E.g., the unit circle in the (Euclidean) plane can be defined algebraically as the zero set of the equation $x^2 + y^2 - 1 = 0$, for $x, y \in \mathbb{R}$.
- Both notions are related:
  - The image of a parametric curve describes an arc.
  - Conversely, an arc admits a parametrization.

Curves

- Intuitively, a curve in $\mathbb{R}^2$ is generated by a continuous motion of a pencil on a sheet of paper.
- A formal mathematical definition is not entirely straightforward, and the term "curve" is associated with two closely related notions: kinematic and geometric.
- In the kinematic setting, a (parametric) curve is a function of one real variable.
- In the geometric setting, a curve, also called an arc, is a 1-dimensional subset of space that is "similar" to a line (albeit it need not be straight).
- E.g., the unit circle in the (Euclidean) plane can be defined algebraically as the zero set of the equation $x^2 + y^2 - 1 = 0$, for $x, y \in \mathbb{R}$.
- Both notions are related:
  - The image of a parametric curve describes an arc.
  - Conversely, an arc admits a parametrization.
- Note that fairly counter-intuitive curves exist: e.g., space-filling curves like the Sierpinski curve. (See next slide.)
- Similarly, the zero set of an algebraic equation in two variables $x, y \in \mathbb{R}$ need not match our intuition of a curve. E.g., $x \cdot y = 0$ models the coordinate axes of $\mathbb{R}^2$.

Caveat: Sierpinski Curves

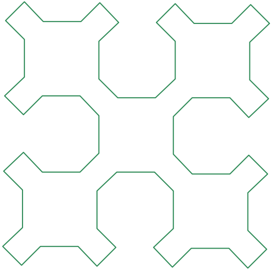
- Sierpinski curves are a sequence of recursively defined continuous and closed curves S_n in $\mathbb{R}^2$.
- Sierpinski curve S_1 of order 1:



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

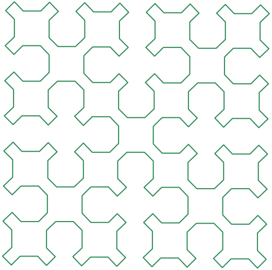
Caveat: Sierpinski Curves

- Sierpinski curves are a sequence of recursively defined continuous and closed curves S_n in $\mathbb{R}^2$.
- Sierpinski curve S_2 of order 2:



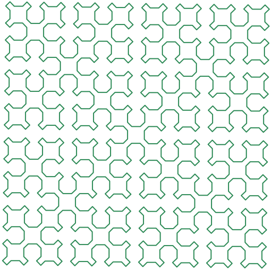
Caveat: Sierpinski Curves

- Sierpinski curves are a sequence of recursively defined continuous and closed curves S_n in $\mathbb{R}^2$.
- Sierpinski curve S_3 of order 3:



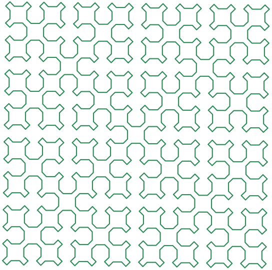
Caveat: Sierpinski Curves

- Sierpinski curves are a sequence of recursively defined continuous and closed curves S_n in $\mathbb{R}^2$.
- Sierpinski curve S_4 of order 4:



Caveat: Sierpinski Curves

- Sierpinski curves are a sequence of recursively defined continuous and closed curves S_n in $\mathbb{R}^2$.
- Sierpinski curve S_4 of order 4:

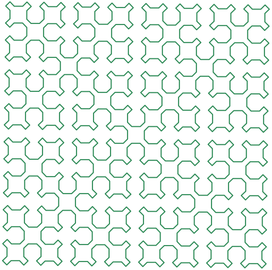


- Their limit curve, *the Sierpinski curve*, is a space-filling curve: In the limit, for $n \to \infty$, it fills the unit square completely!

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Caveat: Sierpinski Curves

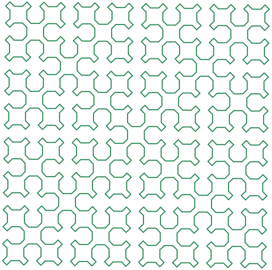
- Sierpinski curves are a sequence of recursively defined continuous and closed curves S_n in $\mathbb{R}^2$.
- Sierpinski curve S_4 of order 4:



- Their limit curve, *the Sierpinski curve*, is a space-filling curve: In the limit, for $n \to \infty$, it fills the unit square completely!
- Its length grows exponentially and unboundedly as n grows.

Caveat: Sierpinski Curves

- Sierpinski curves are a sequence of recursively defined continuous and closed curves S_n in $\mathbb{R}^2$.
- Sierpinski curve S_4 of order 4:



- Their limit curve, *the Sierpinski curve*, is a space-filling curve: In the limit, for $n \to \infty$, it fills the unit square completely!
- Its length grows exponentially and unboundedly as n grows.
- Other space-filling curves exist: E.g., Peano curve, Hilbert curve.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 41 (Curve, Dt.: Kurve)

Let $I \subseteq \mathbb{R}$ be an interval of the real line. A continuous (vector-valued) mapping $\gamma\colon I \to \mathbb{R}^n$ is called a *parametrization* of $\gamma(I)$ or a *parametric curve*.

Curves in $\mathbb{R}^n$

Definition 41 (Curve, Dt.: Kurve)

Let $I \subseteq \mathbb{R}$ be an interval of the real line. A continuous (vector-valued) mapping $\gamma\colon I \to \mathbb{R}^n$ is called a *parametrization* of $\gamma(I)$ or a *parametric curve*.

- Well-known examples of parametric curves include a straight-line segment, a circular arc, and a helix.
- E.g., $\gamma\colon [0,1] \to \mathbb{R}^3$ with

$$\gamma(t) := \begin{pmatrix} p_x + t \cdot (q_x - p_x) \\ p_y + t \cdot (q_y - p_y) \\ p_z + t \cdot (q_z - p_z) \end{pmatrix}$$

maps $[0,1]$ to a straight-line segment from point p to q.

Definition 41 (Curve, Dt.: Kurve)

Let $I \subseteq \mathbb{R}$ be an interval of the real line. A continuous (vector-valued) mapping $\gamma\colon I \to \mathbb{R}^n$ is called a *parametrization* of $\gamma(I)$ or a *parametric curve*.

- Well-known examples of parametric curves include a straight-line segment, a circular arc, and a helix.
- E.g., $\gamma\colon [0,1] \to \mathbb{R}^3$ with

$$\gamma(t) := \begin{pmatrix} p_x + t \cdot (q_x - p_x) \\ p_y + t \cdot (q_y - p_y) \\ p_z + t \cdot (q_z - p_z) \end{pmatrix}$$

  maps $[0,1]$ to a straight-line segment from point p to q.
- The interval I is called the *domain* of γ, and $\gamma(I)$ is called *image* (Dt.: Bild, Spur).

Curves in $\mathbb{R}^n$

Definition 41 (Curve, Dt.: Kurve)

Let $I \subseteq \mathbb{R}$ be an interval of the real line. A continuous (vector-valued) mapping $\gamma\colon I \to \mathbb{R}^n$ is called a *parametrization* of $\gamma(I)$ or a *parametric curve*.

- Well-known examples of parametric curves include a straight-line segment, a circular arc, and a helix.
- E.g., $\gamma\colon [0,1] \to \mathbb{R}^3$ with

$$\gamma(t) := \begin{pmatrix} p_x + t \cdot (q_x - p_x) \\ p_y + t \cdot (q_y - p_y) \\ p_z + t \cdot (q_z - p_z) \end{pmatrix}$$

  maps $[0,1]$ to a straight-line segment from point p to q.
- The interval I is called the *domain* of γ, and $\gamma(I)$ is called *image* (Dt.: Bild, Spur).

Definition 42 (Plane curve, Dt.: ebene Kurve)

For $\gamma\colon I \to \mathbb{R}^n$, the curve $\gamma(I)$ is *plane* if $\gamma(I) \subseteq \mathbb{R}^2$ or if $\gamma(I)$ lies within a plane. A non-plane curve is called a *skew curve* (Dt.: Raumkurve).

Definition 43 (Start and end point)

If I is a closed interval $[a, b]$, for some $a, b \in \mathbb{R}$, then we call $\gamma(a)$ the *start point* and $\gamma(b)$ the *end point* of the curve $\gamma\colon I \to \mathbb{R}^n$.

Definition 43 (Start and end point)

If I is a closed interval $[a, b]$, for some $a, b \in \mathbb{R}$, then we call $\gamma(a)$ the *start point* and $\gamma(b)$ the *end point* of the curve $\gamma\colon I \to \mathbb{R}^n$.

Definition 44 (Closed, Dt.: geschlossen)

A parametrization $\gamma\colon I \to \mathbb{R}^n$ is said to be *closed* (or a *loop*) if I is a closed interval $[a, b]$, for some $a, b \in \mathbb{R}$, and if $\gamma(a) = \gamma(b)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 43 (Start and end point)

If I is a closed interval $[a, b]$, for some $a, b \in \mathbb{R}$, then we call $\gamma(a)$ the *start point* and $\gamma(b)$ the *end point* of the curve $\gamma\colon I \to \mathbb{R}^n$.

Definition 44 (Closed, Dt.: geschlossen)

A parametrization $\gamma\colon I \to \mathbb{R}^n$ is said to be *closed* (or a *loop*) if I is a closed interval $[a, b]$, for some $a, b \in \mathbb{R}$, and if $\gamma(a) = \gamma(b)$.

Definition 45 (Simple, Dt.: einfach)

A parametrization $\gamma\colon I \to \mathbb{R}^n$ is said to be *simple* if $\gamma(t_1) = \gamma(t_2)$ for $t_1 \neq t_2 \in I$ implies $I = [a, b]$ for some $a, b \in \mathbb{R}$ and $\{t_1, t_2\} = \{a, b\}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 43 (Start and end point)

If I is a closed interval $[a, b]$, for some $a, b \in \mathbb{R}$, then we call $\gamma(a)$ the *start point* and $\gamma(b)$ the *end point* of the curve $\gamma\colon I \to \mathbb{R}^n$.

Definition 44 (Closed, Dt.: geschlossen)

A parametrization $\gamma\colon I \to \mathbb{R}^n$ is said to be *closed* (or a *loop*) if I is a closed interval $[a, b]$, for some $a, b \in \mathbb{R}$, and if $\gamma(a) = \gamma(b)$.

Definition 45 (Simple, Dt.: einfach)

A parametrization $\gamma\colon I \to \mathbb{R}^n$ is said to be *simple* if $\gamma(t_1) = \gamma(t_2)$ for $t_1 \neq t_2 \in I$ implies $I = [a, b]$ for some $a, b \in \mathbb{R}$ and $\{t_1, t_2\} = \{a, b\}$.

- Hence, if $\gamma\colon I \to \mathbb{R}^n$ is simple then it is injective on $\mathtt{int}(I)$: It has no "self-intersections".

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Many properties of curves can also be stated independently of a specific parametrization. E.g., we can regard a curve $\mathcal{C}$ to be simple if there exists one parametrization of $\mathcal{C}$ that is simple.

Curves in $\mathbb{R}^n$

- Many properties of curves can also be stated independently of a specific parametrization. E.g., we can regard a curve $\mathcal{C}$ to be simple if there exists one parametrization of $\mathcal{C}$ that is simple.
- In daily math, the standard meaning of a “curve” is the image of the equivalence class of all paths under a certain equivalence relation. (Roughly, two paths are equivalent if they are identical up to re-parametrization.)
- Hence, the distinction between a curve and (one of) its parametrizations is often blurred.

Curves in $\mathbb{R}^n$

- Many properties of curves can also be stated independently of a specific parametrization. E.g., we can regard a curve $\mathcal{C}$ to be simple if there exists one parametrization of $\mathcal{C}$ that is simple.
- In daily math, the standard meaning of a “curve” is the image of the equivalence class of all paths under a certain equivalence relation. (Roughly, two paths are equivalent if they are identical up to re-parametrization.)
- Hence, the distinction between a curve and (one of) its parametrizations is often blurred.
- For the sake of simplicity, we will not distinguish between a curve $\mathcal{C}$ and one of its parametrizations γ if the meaning is clear.
- Similarly, we will frequently call γ a curve.
- For instance, we will frequently speak about a closed curve rather than about a closed parametrization of a curve.

Convex Curve in $\mathbb{R}^2$

Definition 46 (Supporting line, Dt.: Stützgerade)

In $\mathbb{R}^2$, a line ℓ is a supporting line of a curve $\mathcal{C}$ if

1. ℓ passes through a point of $\mathcal{C}$,
2. $\mathcal{C}$ lies completely in one of the two closed half-planes induced by ℓ.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 46 (Supporting line, Dt.: Stützgerade)

In $\mathbb{R}^2$, a line ℓ is a supporting line of a curve $\mathcal{C}$ if

1. ℓ passes through a point of $\mathcal{C}$,
2. $\mathcal{C}$ lies completely in one of the two closed half-planes induced by ℓ.

- There may be many supporting lines for a curve at a given point.

Definition 46 (Supporting line, Dt.: Stützgerade)

In $\mathbb{R}^2$, a line ℓ is a supporting line of a curve $\mathcal{C}$ if

1. ℓ passes through a point of $\mathcal{C}$,
2. $\mathcal{C}$ lies completely in one of the two closed half-planes induced by ℓ.

- There may be many supporting lines for a curve at a given point.
- If a tangent exists at a given point, then it is the unique supporting line at this point if it does not separate the curve.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 46 (Supporting line, Dt.: Stützgerade)

In $\mathbb{R}^2$, a line ℓ is a supporting line of a curve $\mathcal{C}$ if

1. ℓ passes through a point of $\mathcal{C}$,
2. $\mathcal{C}$ lies completely in one of the two closed half-planes induced by ℓ.

- There may be many supporting lines for a curve at a given point.
- If a tangent exists at a given point, then it is the unique supporting line at this point if it does not separate the curve.

Definition 47 (Convex curve)

In $\mathbb{R}^2$, a curve is convex if it has a supporting line through each of its points.

Convex Curve in $\mathbb{R}^2$

Definition 46 (Supporting line, Dt.: Stützgerade)

In $\mathbb{R}^2$, a line ℓ is a supporting line of a curve $\mathcal{C}$ if

1. ℓ passes through a point of $\mathcal{C}$,
2. $\mathcal{C}$ lies completely in one of the two closed half-planes induced by ℓ.

- There may be many supporting lines for a curve at a given point.
- If a tangent exists at a given point, then it is the unique supporting line at this point if it does not separate the curve.

Definition 47 (Convex curve)

In $\mathbb{R}^2$, a curve is convex if it has a supporting line through each of its points.

Lemma 48

Every convex curve is a subset of the boundary of its own convex hull.

Definition 46 (Supporting line, Dt.: Stützgerade)

In $\mathbb{R}^2$, a line ℓ is a supporting line of a curve $\mathcal{C}$ if

1. ℓ passes through a point of $\mathcal{C}$,
2. $\mathcal{C}$ lies completely in one of the two closed half-planes induced by ℓ.

- There may be many supporting lines for a curve at a given point.
- If a tangent exists at a given point, then it is the unique supporting line at this point if it does not separate the curve.

Definition 47 (Convex curve)

In $\mathbb{R}^2$, a curve is convex if it has a supporting line through each of its points.

Lemma 48

Every convex curve is a subset of the boundary of its own convex hull.

- It is straightforward to extend the notion of convexity from $\mathbb{R}^2$ to plane curves.

Jordan Curve in $\mathbb{R}^2$

Definition 49 (Jordan curve)

A set $\mathcal{C} \subset \mathbb{R}^2$ (which is not a single point) is called a *Jordan curve* if there exists a simple and closed parametrization $\gamma : I \to \mathbb{R}^2$ that parameterizes $\mathcal{C}$.

Jordan Curve in $\mathbb{R}^2$

Definition 49 (Jordan curve)

A set $\mathcal{C} \subset \mathbb{R}^2$ (which is not a single point) is called a *Jordan curve* if there exists a simple and closed parametrization $\gamma : I \to \mathbb{R}^2$ that parameterizes $\mathcal{C}$.

Theorem 50 (Jordan 1887)

Every Jordan curve $\mathcal{C}$ partitions $\mathbb{R}^2 \setminus \mathcal{C}$ into two disjoint open regions, a (bounded) "interior" region and an (unbounded) "exterior" region, with $\mathcal{C}$ as the (topological) boundary of both of them.

Definition 49 (Jordan curve)

A set $\mathcal{C} \subset \mathbb{R}^2$ (which is not a single point) is called a *Jordan curve* if there exists a simple and closed parametrization $\gamma : I \rightarrow \mathbb{R}^2$ that parameterizes $\mathcal{C}$.

Theorem 50 (Jordan 1887)

Every Jordan curve $\mathcal{C}$ partitions $\mathbb{R}^2 \setminus \mathcal{C}$ into two disjoint open regions, a (bounded) "interior" region and an (unbounded) "exterior" region, with $\mathcal{C}$ as the (topological) boundary of both of them.

- Although this theorem — the so-called Jordan Curve Theorem (Dt.: Jordanscher Kurvensatz) — seems obvious, a proof is not entirely trivial.

Jordan Curve in $\mathbb{R}^2$

Definition 49 (Jordan curve)

A set $\mathcal{C} \subset \mathbb{R}^2$ (which is not a single point) is called a *Jordan curve* if there exists a simple and closed parametrization $\gamma : I \to \mathbb{R}^2$ that parameterizes $\mathcal{C}$.

Theorem 50 (Jordan 1887)

Every Jordan curve $\mathcal{C}$ partitions $\mathbb{R}^2 \setminus \mathcal{C}$ into two disjoint open regions, a (bounded) "interior" region and an (unbounded) "exterior" region, with $\mathcal{C}$ as the (topological) boundary of both of them.

- Although this theorem — the so-called Jordan Curve Theorem (Dt.: Jordanscher Kurvensatz) — seems obvious, a proof is not entirely trivial.

Theorem 51 (Schönflies 1906)

For every Jordan curve $\mathcal{C}$ there exists a homeomorphism from the plane to itself that maps $\mathcal{C}$ to the unit sphere S^1.

- Roughly, a homeomorphism is a bijective continuous stretching and bending of one space into another space such that the inverse function also is continuous.

Definition 52 (C^r-parametrization)

If $\gamma\colon I \to \mathbb{R}^n$ is r times continuously differentiable then γ is called a parametric curve of class C^r, or a C^r*-parametrization* of $\gamma(I)$, or simply a C^r-curve.

Definition 52 (C^r-parametrization)

If $\gamma\colon I \to \mathbb{R}^n$ is r times continuously differentiable then γ is called a parametric curve of class C^r, or a *C^r-parametrization* of $\gamma(I)$, or simply a C^r-curve.
If $I = [a, b]$, then γ is called a *closed C^r-parametrization* if $\gamma^{(k)}(a) = \gamma^{(k)}(b)$ for all $0 \leq k \leq r$.

- One-sided differentiability is meant at the endpoints of I if I is a closed interval.

Definition 52 (C^r-parametrization)

If $\gamma\colon I \to \mathbb{R}^n$ is r times continuously differentiable then γ is called a parametric curve of class C^r, or a *C^r-parametrization* of $\gamma(I)$, or simply a C^r-curve.
If $I = [a, b]$, then γ is called a *closed C^r-parametrization* if $\gamma^{(k)}(a) = \gamma^{(k)}(b)$ for all $0 \leq k \leq r$.

- One-sided differentiability is meant at the endpoints of I if I is a closed interval.

Definition 53 (Smooth curve, Dt.: glatte Kurve)

If $\gamma\colon I \to \mathbb{R}^n$ has derivatives of all orders then γ is (the parametrization of) a *smooth curve* (or of class C^∞).

Definition 52 (C^r-parametrization)

If $\gamma\colon I \to \mathbb{R}^n$ is r times continuously differentiable then γ is called a parametric curve of class C^r, or a *C^r-parametrization* of $\gamma(I)$, or simply a C^r-curve.
If $I = [a, b]$, then γ is called a *closed C^r-parametrization* if $\gamma^{(k)}(a) = \gamma^{(k)}(b)$ for all $0 \le k \le r$.

- One-sided differentiability is meant at the endpoints of I if I is a closed interval.

Definition 53 (Smooth curve, Dt.: glatte Kurve)

If $\gamma\colon I \to \mathbb{R}^n$ has derivatives of all orders then γ is (the parametrization of) a *smooth curve* (or of class C^∞).

Definition 54 (Piecewise smooth curve, Dt.: stückweise glatte Kurve)

If I is the union of a finite number of sub-intervals over each of which $\gamma\colon I \to \mathbb{R}^n$ is smooth then γ is *piecewise smooth*.

Definition 52 (C^r-parametrization)

If $\gamma\colon I \to \mathbb{R}^n$ is r times continuously differentiable then γ is called a parametric curve of class C^r, or a *C^r-parametrization* of $\gamma(I)$, or simply a C^r-curve.
If $I = [a, b]$, then γ is called a *closed C^r-parametrization* if $\gamma^{(k)}(a) = \gamma^{(k)}(b)$ for all $0 \leq k \leq r$.

- One-sided differentiability is meant at the endpoints of I if I is a closed interval.

Definition 53 (Smooth curve, Dt.: glatte Kurve)

If $\gamma\colon I \to \mathbb{R}^n$ has derivatives of all orders then γ is (the parametrization of) a *smooth curve* (or of class C^∞).

Definition 54 (Piecewise smooth curve, Dt.: stückweise glatte Kurve)

If I is the union of a finite number of sub-intervals over each of which $\gamma\colon I \to \mathbb{R}^n$ is smooth then γ is *piecewise smooth*.

- Smoothness depends on the parametrization!

Definition 52 (C^r-parametrization)

If $\gamma\colon I \to \mathbb{R}^n$ is r times continuously differentiable then γ is called a parametric curve of class C^r, or a C^r*-parametrization* of $\gamma(I)$, or simply a C^r-curve.
If $I = [a, b]$, then γ is called a *closed* C^r*-parametrization* if $\gamma^{(k)}(a) = \gamma^{(k)}(b)$ for all $0 \le k \le r$.

- One-sided differentiability is meant at the endpoints of I if I is a closed interval.

Definition 53 (Smooth curve, Dt.: glatte Kurve)

If $\gamma\colon I \to \mathbb{R}^n$ has derivatives of all orders then γ is (the parametrization of) a *smooth curve* (or of class C^∞).

Definition 54 (Piecewise smooth curve, Dt.: stückweise glatte Kurve)

If I is the union of a finite number of sub-intervals over each of which $\gamma\colon I \to \mathbb{R}^n$ is smooth then γ is *piecewise smooth*.

- Smoothness depends on the parametrization!
- [Weierstrass (1872), Koch (1904)]: There do exist curves which are continuous everywhere but differentiable nowhere.

Definition 55 (Regular, Dt.: regulär)

A C^r-curve $\gamma\colon I \to \mathbb{R}^n$ is called *regular of order k*, for some $0 < k \leq r$, if the vectors $\{\gamma'(t), \gamma''(t), \ldots, \gamma^{(k)}(t)\}$ are linearly independent for every $t \in I$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 55 (Regular, Dt.: regulär)

A C^r-curve $\gamma\colon I \to \mathbb{R}^n$ is called *regular of order* k, for some $0 < k \leq r$, if the vectors $\{\gamma'(t), \gamma''(t), \ldots, \gamma^{(k)}(t)\}$ are linearly independent for every $t \in I$.
In particular, γ is called *regular* if $\gamma'(t) \neq 0 \in \mathbb{R}^n$ for every $t \in I$.

Definition 55 (Regular, Dt.: regulär)

A C^r-curve $\gamma\colon I \to \mathbb{R}^n$ is called *regular of order k*, for some $0 < k \leq r$, if the vectors $\{\gamma'(t), \gamma''(t), \ldots, \gamma^{(k)}(t)\}$ are linearly independent for every $t \in I$.
In particular, γ is called *regular* if $\gamma'(t) \neq 0 \in \mathbb{R}^n$ for every $t \in I$.

Definition 56 (Singular, Dt.: singulär)

For a C^1-curve $\gamma\colon I \to \mathbb{R}^n$ and $t_0 \in I$, the point $\gamma(t_0)$ is called a *singular point* of γ if $\gamma'(t_0) = 0$.

- Regularity and singularity depend on the parametrization!

Equivalence of Parametrizations in $\mathbb{R}^n$

- Parametrizations of a curve (regarded as a set $\mathcal{C} \subset \mathbb{R}^n$) need not be unique: Two different parametrizations $\gamma\colon I \to \mathbb{R}^n$ and $\beta\colon J \to \mathbb{R}^n$ may exist such that $\mathcal{C} = \gamma(I) = \beta(J)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Parametrizations of a curve (regarded as a set $\mathcal{C} \subset \mathbb{R}^n$) need not be unique: Two different parametrizations $\gamma\colon I \to \mathbb{R}^n$ and $\beta\colon J \to \mathbb{R}^n$ may exist such that $\mathcal{C} = \gamma(I) = \beta(J)$.

$$\gamma(t) := \begin{pmatrix} \cos 2\pi\, t \\ \sin 2\pi\, t \end{pmatrix} \qquad \beta(t) := \begin{pmatrix} \cos 2\pi\, t \\ -\sin 2\pi\, t \end{pmatrix}$$

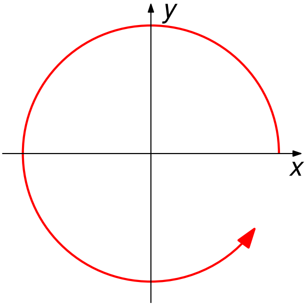


Figure: $\gamma(t)$ for $t \in [0, 0.9]$

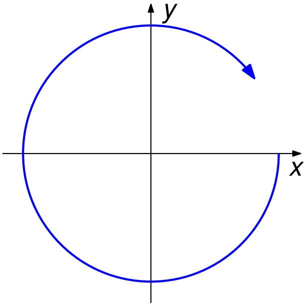


Figure: $\beta(t)$ for $t \in [0, 0.9]$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 57 (Reparametrization, Dt.: Umparameterisierung)

Let $\gamma\colon I \to \mathbb{R}^n$ and $\beta\colon J \to \mathbb{R}^n$ both be C^r-curves, for some $r \in \mathbb{N}_0$. We consider γ and β as *equivalent* if a function $\phi\colon I \to J$ exists, such that

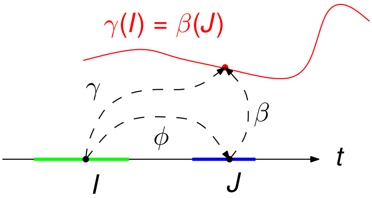
$$\beta(\phi(t)) = \gamma(t) \qquad \forall t \in I,$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 57 (Reparametrization, Dt.: Umparameterisierung)

Let $\gamma\colon I \to \mathbb{R}^n$ and $\beta\colon J \to \mathbb{R}^n$ both be C^r-curves, for some $r \in \mathbb{N}_0$. We consider γ and β as *equivalent* if a function $\phi\colon I \to J$ exists, such that

$$\beta(\phi(t)) = \gamma(t) \qquad \forall t \in I,$$



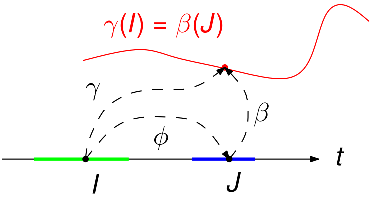
Definition 57 (Reparametrization, Dt.: Umparameterisierung)

Let $\gamma\colon I \to \mathbb{R}^n$ and $\beta\colon J \to \mathbb{R}^n$ both be C^r-curves, for some $r \in \mathbb{N}_0$. We consider γ and β as *equivalent* if a function $\phi\colon I \to J$ exists, such that

$$\beta(\phi(t)) = \gamma(t) \qquad \forall t \in I,$$

and

1. ϕ is continuous, strictly monotonously increasing and bijective,



Definition 57 (Reparametrization, Dt.: Umparameterisierung)

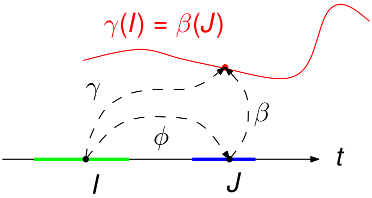
Let $\gamma\colon I \to \mathbb{R}^n$ and $\beta\colon J \to \mathbb{R}^n$ both be C^r-curves, for some $r \in \mathbb{N}_0$. We consider γ and β as *equivalent* if a function $\phi\colon I \to J$ exists, such that

$$\beta(\phi(t)) = \gamma(t) \qquad \forall t \in I,$$

and

1. ϕ is continuous, strictly monotonously increasing and bijective,
2. both ϕ and ϕ^{-1} are r times continuously differentiable.

In this case the parametric curve β is called a *reparametrization* of γ.



Equivalence of Parametrizations in $\mathbb{R}^n$

Definition 57 (Reparametrization, Dt.: Umparameterisierung)

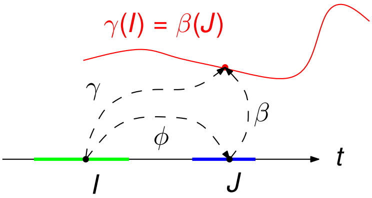
Let $\gamma\colon I \to \mathbb{R}^n$ and $\beta\colon J \to \mathbb{R}^n$ both be C^r-curves, for some $r \in \mathbb{N}_0$. We consider γ and β as *equivalent* if a function $\phi\colon I \to J$ exists, such that

$$\beta(\phi(t)) = \gamma(t) \qquad \forall t \in I,$$

and

1. ϕ is continuous, strictly monotonously increasing and bijective,
2. both ϕ and ϕ^{-1} are r times continuously differentiable.

In this case the parametric curve β is called a *reparametrization* of γ.



Caveat

There is no universally accepted definition of a reparametrization! Some authors drop the monotonicity or the differentiability of ϕ, while others even require ϕ to be smooth.

Definition 58 (Decomposition, Dt.: Unterteilung)

Consider $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$. A *decomposition*, D, of the closed interval I is a sequence of $m+1$ numbers $t_0, t_1, t_2, \ldots, t_m$, for some $m \in \mathbb{N}$, such that

$$a = t_0 < t_1 < t_2 < \cdots < t_m = b.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Arc Length

Definition 58 (Decomposition, Dt.: Unterteilung)

Consider $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$. A *decomposition*, D, of the closed interval I is a sequence of $m + 1$ numbers $t_0, t_1, t_2, \ldots, t_m$, for some $m \in \mathbb{N}$, such that

$$a = t_0 < t_1 < t_2 < \cdots < t_m = b.$$

The length $L_D(\gamma)$ of the polygonal chain $(\gamma(t_0), \gamma(t_1), \gamma(t_2), \ldots, \gamma(t_m))$ that corresponds to the decomposition $t_0, t_1, t_2, \ldots, t_m$ is given by

$$L_D(\gamma) := \sum_{j=0}^{m-1} \|\gamma(t_{j+1}) - \gamma(t_j)\|$$

$$= \|\gamma(t_1) - \gamma(t_0)\| + \|\gamma(t_2) - \gamma(t_1)\| + \cdots + \|\gamma(t_m) - \gamma(t_{m-1})\|.$$

Definition 58 (Decomposition, Dt.: Unterteilung)

Consider $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$. A *decomposition*, D, of the closed interval I is a sequence of $m + 1$ numbers $t_0, t_1, t_2, \ldots, t_m$, for some $m \in \mathbb{N}$, such that

$$a = t_0 < t_1 < t_2 < \cdots < t_m = b.$$

The length $L_D(\gamma)$ of the polygonal chain $(\gamma(t_0), \gamma(t_1), \gamma(t_2), \ldots, \gamma(t_m))$ that corresponds to the decomposition $t_0, t_1, t_2, \ldots, t_m$ is given by

$$\begin{aligned} L_D(\gamma) &:= \sum_{j=0}^{m-1} \|\gamma(t_{j+1}) - \gamma(t_j)\| \\ &= \|\gamma(t_1) - \gamma(t_0)\| + \|\gamma(t_2) - \gamma(t_1)\| + \cdots + \|\gamma(t_m) - \gamma(t_{m-1})\|. \end{aligned}$$

We denote the set of all decompositions of $[a, b]$ by $\mathcal{D}[a, b]$.

Definition 59 (Arc length, Dt.: Bogenlänge)

Consider $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$. The *arc length* of $\gamma(I)$ is given by

$$\sup \{L_D(\gamma) : D \in \mathcal{D}[a, b]\},$$

i.e., by the supremum (over all decompositions $t_0, t_1, t_2, \ldots, t_m$ of I) of the lengths of the polygonal chains defined by $\gamma(t_0), \gamma(t_1), \gamma(t_2), \ldots, \gamma(t_m)$.

Arc Length

Definition 59 (Arc length, Dt.: Bogenlänge)

Consider $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$. The *arc length* of $\gamma(I)$ is given by

$$\sup\left\{L_D(\gamma) : D \in \mathcal{D}[a,b]\right\},$$

i.e., by the supremum (over all decompositions $t_0, t_1, t_2, \ldots, t_m$ of I) of the lengths of the polygonal chains defined by $\gamma(t_0), \gamma(t_1), \gamma(t_2), \ldots, \gamma(t_m)$.

Definition 60 (Rectifiable, Dt.: rektifizierbar)

If the arc length of $\gamma\colon I \to \mathbb{R}^n$ is a finite number then $\gamma(I)$ is called *rectifiable*.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Arc Length

Definition 59 (Arc length, Dt.: Bogenlänge)

Consider $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$. The *arc length* of $\gamma(I)$ is given by

$$\sup\left\{L_D(\gamma) : D \in \mathcal{D}[a, b]\right\},$$

i.e., by the supremum (over all decompositions $t_0, t_1, t_2, \ldots, t_m$ of I) of the lengths of the polygonal chains defined by $\gamma(t_0), \gamma(t_1), \gamma(t_2), \ldots, \gamma(t_m)$.

Definition 60 (Rectifiable, Dt.: rektifizierbar)

If the arc length of $\gamma\colon I \to \mathbb{R}^n$ is a finite number then $\gamma(I)$ is called *rectifiable*.

Lemma 61

The arc length of a curve does not change for equivalent parametrizations.

Definition 59 (Arc length, Dt.: Bogenlänge)

Consider $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$. The *arc length* of $\gamma(I)$ is given by

$$\sup\{L_D(\gamma) : D \in \mathcal{D}[a, b]\},$$

i.e., by the supremum (over all decompositions $t_0, t_1, t_2, \ldots, t_m$ of I) of the lengths of the polygonal chains defined by $\gamma(t_0), \gamma(t_1), \gamma(t_2), \ldots, \gamma(t_m)$.

Definition 60 (Rectifiable, Dt.: rektifizierbar)

If the arc length of $\gamma\colon I \to \mathbb{R}^n$ is a finite number then $\gamma(I)$ is called *rectifiable*.

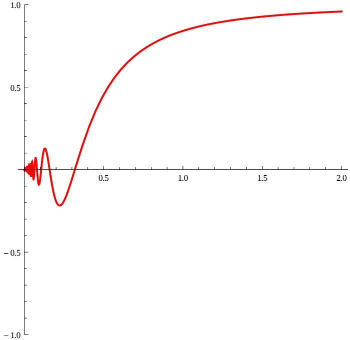
Lemma 61

The arc length of a curve does not change for equivalent parametrizations.

Sketch of proof : Suppose that $\gamma(t) = \beta(\phi(t))$ for all $t \in I$, for $\beta\colon J \to \mathbb{R}^n$. Every decomposition $t_0, t_1, t_2, \ldots, t_m$ of I maps to a decomposition $\phi(t_0),\phi(t_1),\phi(t_2), \ldots, \phi(t_m)$ of J such that $\gamma(t_i) = \beta(\phi(t_i))$ for all $1 \le i \le m$. Hence, there is a bijection from the set of decompositions of I to the set of decompositions of J, and it does not matter which set is used for determining the supremum of all possible chain lengths.

Arc Length: Non-Rectifiable Curve

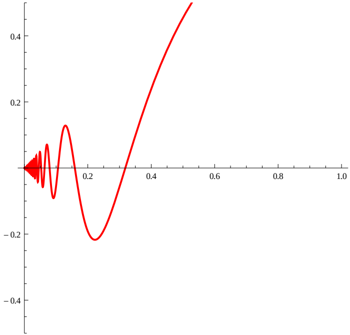
- Curves exist that are non-rectifiable, i.e., for which there is no upper bound on the length of their polygonal approximations.
- Example of a non-rectifiable curve: The graph of the function defined by $f(0) := 0$ and $f(x) := x \sin\left(\frac{1}{x}\right)$ for $0 < x \leq a$, for some $a \in \mathbb{R}^+$. It defines a curve $\gamma(t) := \begin{pmatrix} t \\ f(t) \end{pmatrix}$.



The graph of $f(x) := x \sin\left(\frac{1}{x}\right)$ for $x \in [0, 2]$

Arc Length: Non-Rectifiable Curve

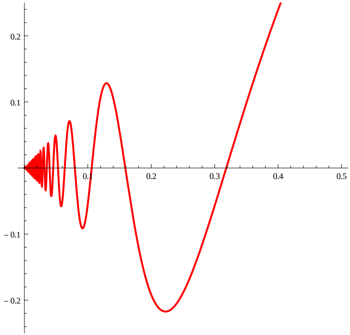
- Curves exist that are non-rectifiable, i.e., for which there is no upper bound on the length of their polygonal approximations.
- Example of a non-rectifiable curve: The graph of the function defined by $f(0) := 0$ and $f(x) := x \sin\left(\frac{1}{x}\right)$ for $0 < x \leq a$, for some $a \in \mathbb{R}^+$. It defines a curve $\gamma(t) := \begin{pmatrix} t \\ f(t) \end{pmatrix}$.



The graph of $f(x) := x \sin\left(\frac{1}{x}\right)$ for $x \in [0, 1]$

Arc Length: Non-Rectifiable Curve

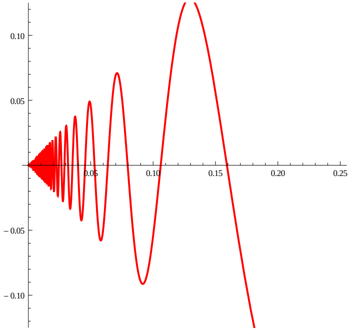
- Curves exist that are non-rectifiable, i.e., for which there is no upper bound on the length of their polygonal approximations.
- Example of a non-rectifiable curve: The graph of the function defined by $f(0) := 0$ and $f(x) := x \sin\left(\frac{1}{x}\right)$ for $0 < x \leq a$, for some $a \in \mathbb{R}^+$. It defines a curve $\gamma(t) := \begin{pmatrix} t \\ f(t) \end{pmatrix}$.



The graph of $f(x) := x \sin\left(\frac{1}{x}\right)$ for $x \in [0, \frac{1}{2}]$

Arc Length: Non-Rectifiable Curve

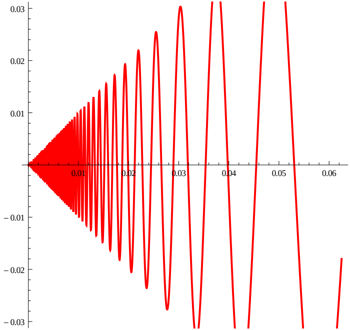
- Curves exist that are non-rectifiable, i.e., for which there is no upper bound on the length of their polygonal approximations.
- Example of a non-rectifiable curve: The graph of the function defined by $f(0) := 0$ and $f(x) := x \sin\left(\frac{1}{x}\right)$ for $0 < x \leq a$, for some $a \in \mathbb{R}^+$. It defines a curve $\gamma(t) := \begin{pmatrix} t \\ f(t) \end{pmatrix}$.



The graph of $f(x) := x \sin\left(\frac{1}{x}\right)$ for $x \in [0, \frac{1}{4}]$

Arc Length: Non-Rectifiable Curve

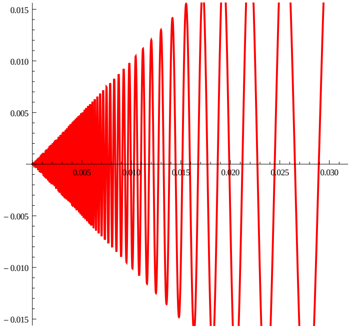
- Curves exist that are non-rectifiable, i.e., for which there is no upper bound on the length of their polygonal approximations.
- Example of a non-rectifiable curve: The graph of the function defined by $f(0) := 0$ and $f(x) := x \sin\left(\frac{1}{x}\right)$ for $0 < x \leq a$, for some $a \in \mathbb{R}^+$. It defines a curve $\gamma(t) := \begin{pmatrix} t \\ f(t) \end{pmatrix}$.



The graph of $f(x) := x \sin\left(\frac{1}{x}\right)$ for $x \in [0, \frac{1}{8}]$

Arc Length: Non-Rectifiable Curve

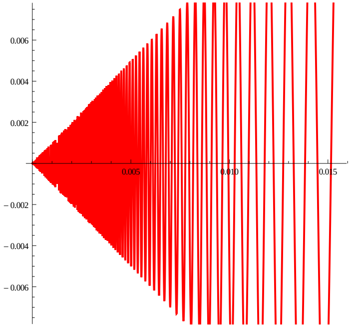
- Curves exist that are non-rectifiable, i.e., for which there is no upper bound on the length of their polygonal approximations.
- Example of a non-rectifiable curve: The graph of the function defined by $f(0) := 0$ and $f(x) := x \sin\left(\frac{1}{x}\right)$ for $0 < x \leq a$, for some $a \in \mathbb{R}^+$. It defines a curve $\gamma(t) := \begin{pmatrix} t \\ f(t) \end{pmatrix}$.



The graph of $f(x) := x \sin\left(\frac{1}{x}\right)$ for $x \in [0, \frac{1}{16}]$

Arc Length: Non-Rectifiable Curve

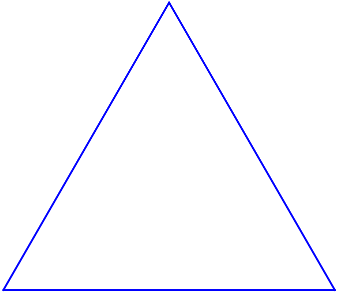
- Curves exist that are non-rectifiable, i.e., for which there is no upper bound on the length of their polygonal approximations.
- Example of a non-rectifiable curve: The graph of the function defined by $f(0) := 0$ and $f(x) := x \sin\left(\frac{1}{x}\right)$ for $0 < x \leq a$, for some $a \in \mathbb{R}^+$. It defines a curve $\gamma(t) := \begin{pmatrix} t \\ f(t) \end{pmatrix}$.



The graph of $f(x) := x \sin\left(\frac{1}{x}\right)$ for $x \in [0, \frac{1}{32}]$

Arc Length: Non-Rectifiable Curve

- Example of a non-rectifiable closed curve: The *Koch snowflake* [Koch 1904].

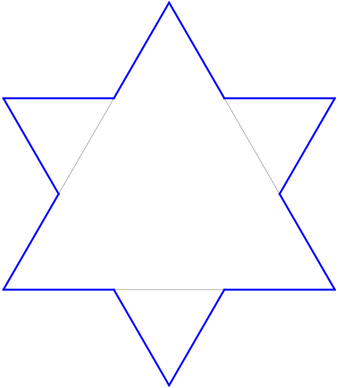


Koch snowflake, iteration 0

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Arc Length: Non-Rectifiable Curve

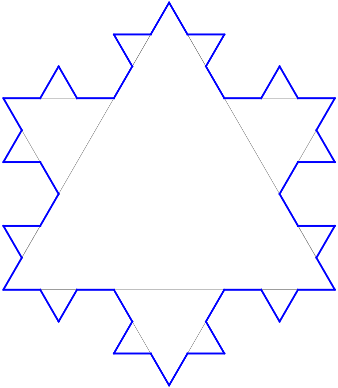
- Example of a non-rectifiable closed curve: The *Koch snowflake* [Koch 1904].



Koch snowflake, iteration 1

Arc Length: Non-Rectifiable Curve

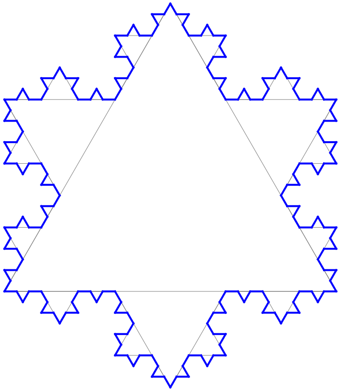
- Example of a non-rectifiable closed curve: The *Koch snowflake* [Koch 1904].



Koch snowflake, iteration 2

Arc Length: Non-Rectifiable Curve

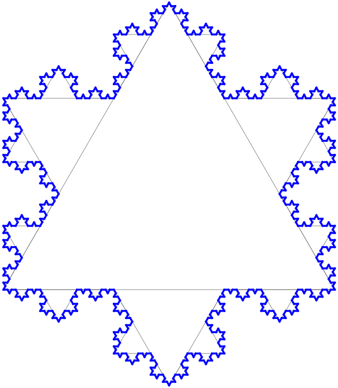
- Example of a non-rectifiable closed curve: The *Koch snowflake* [Koch 1904].



Koch snowflake, iteration 3

Arc Length: Non-Rectifiable Curve

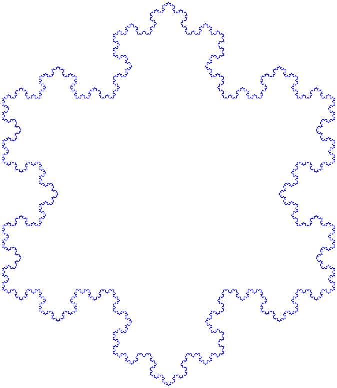
- Example of a non-rectifiable closed curve: The *Koch snowflake* [Koch 1904].



Koch snowflake, iteration 4

Arc Length: Non-Rectifiable Curve

- Example of a non-rectifiable closed curve: The *Koch snowflake* [Koch 1904].



Koch snowflake, iteration 5

- The length of the curve after the n-th iteration is $(^4/_3)^n$ times the original triangle perimeter. (Its fractal dimension is $^{\log 4}/_{\log 3} \approx 1.261$.)

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Theorem 62

If $\gamma\colon I \to \mathbb{R}^n$ is a C^1-curve then $\gamma(I)$ is rectifiable.

Theorem 62

If $\gamma\colon I \to \mathbb{R}^n$ is a C^1-curve then $\gamma(I)$ is rectifiable.

Theorem 63

Let $\gamma\colon I \to \mathbb{R}^n$ be a C^1 curve, with $I := [a, b]$. Then the arc length of $\gamma(I)$ is given by

$$\int_a^b \|\gamma'(t)\| \; dt.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Theorem 62

If $\gamma\colon I \to \mathbb{R}^n$ is a C^1-curve then $\gamma(I)$ is rectifiable.

Theorem 63

Let $\gamma\colon I \to \mathbb{R}^n$ be a C^1 curve, with $I := [a, b]$. Then the arc length of $\gamma(I)$ is given by

$$\int_a^b \|\gamma'(t)\| \, dt.$$

Corollary 64

Let $\gamma\colon I \to \mathbb{R}^n$ be a C^1 curve, and $[a, b] \subseteq I$. Then the arc length of $\gamma([a, b])$ is given by

$$\int_a^b \|\gamma'(t)\| \, dt.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Arc Length: Unit Speed

Definition 65 (Speed, Dt.: Geschwindigkeit)

If $\gamma\colon I \to \mathbb{R}^n$ is a C^1-curve then the vector $\gamma'(t)$ is the *velocity vector* at parameter t, and $\|\gamma'(t)\|$ gives the *speed* at parameter t, for all $t \in I$.

Definition 65 (Speed, Dt.: Geschwindigkeit)

If $\gamma\colon I \to \mathbb{R}^n$ is a C^1-curve then the vector $\gamma'(t)$ is the *velocity vector* at parameter t, and $\|\gamma'(t)\|$ gives the *speed* at parameter t, for all $t \in I$.

Definition 66 (Natural parametrization)

A C^1-curve $\gamma\colon I \to \mathbb{R}^n$ is called *natural* (or at *unit speed*) if $\|\gamma'(t)\| = 1$ for all $t \in I$.

Arc Length: Unit Speed

Definition 65 (Speed, Dt.: Geschwindigkeit)

If $\gamma\colon I \to \mathbb{R}^n$ is a C^1-curve then the vector $\gamma'(t)$ is the *velocity vector* at parameter t, and $\|\gamma'(t)\|$ gives the *speed* at parameter t, for all $t \in I$.

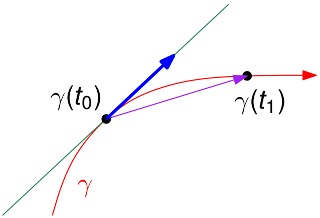
Definition 66 (Natural parametrization)

A C^1-curve $\gamma\colon I \to \mathbb{R}^n$ is called *natural* (or at *unit speed*) if $\|\gamma'(t)\| = 1$ for all $t \in I$.

Theorem 67

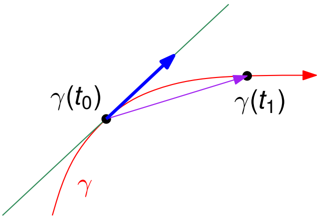
If $\gamma\colon I \to \mathbb{R}^n$, with $I := [a, b]$, is a regular curve then there exists an equivalent reparametrization $\tilde{\gamma}$ that has unit speed.

Tangent Vector



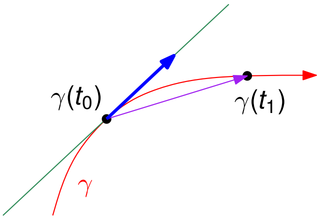
- If $\gamma(t_0)$ is a fixed point on the curve γ, and $\gamma(t_1)$, with $t_1 > t_0$, is another point, then the vector from $\gamma(t_0)$ to $\gamma(t_1)$ approaches the *tangent vector* to γ at $\gamma(t_0)$ as the distance between t_1 and t_0 is decreased.

Tangent Vector



- If $\gamma(t_0)$ is a fixed point on the curve γ, and $\gamma(t_1)$, with $t_1 > t_0$, is another point, then the vector from $\gamma(t_0)$ to $\gamma(t_1)$ approaches the *tangent vector* to γ at $\gamma(t_0)$ as the distance between t_1 and t_0 is decreased.
- The infinite line through $\gamma(t_0)$ that is parallel to this vector is known as the *tangent line* to the curve γ at point $\gamma(t_0)$.

Tangent Vector



- If $\gamma(t_0)$ is a fixed point on the curve γ, and $\gamma(t_1)$, with $t_1 > t_0$, is another point, then the vector from $\gamma(t_0)$ to $\gamma(t_1)$ approaches the *tangent vector* to γ at $\gamma(t_0)$ as the distance between t_1 and t_0 is decreased.
- The infinite line through $\gamma(t_0)$ that is parallel to this vector is known as the *tangent line* to the curve γ at point $\gamma(t_0)$.
- If we disregard the orientation of the tangent vector then we would like to obtain the same result for the tangent line by considering a point $\gamma(t_1)$ with $t_1 < t_0$.

Definition 68 (Tangent vector)

Let $\gamma\colon I \to \mathbb{R}^n$ be a C^1-curve. If $\gamma'(t) \neq 0$ for $t \in I$ then $\gamma'(t)$ forms the *tangent vector* at the point $\gamma(t)$ of γ.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Tangent Vector

Definition 68 (Tangent vector)

Let $\gamma\colon I \to \mathbb{R}^n$ be a C^1-curve. If $\gamma'(t) \neq 0$ for $t \in I$ then $\gamma'(t)$ forms the *tangent vector* at the point $\gamma(t)$ of γ.

- The tangent vector indicates the forward direction of γ relative to increasing parameter values.
- If γ is at unit speed then $\gamma'(t)$ forms a unit vector.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 68 (Tangent vector)

Let $\gamma\colon I \to \mathbb{R}^n$ be a C^1-curve. If $\gamma'(t) \neq 0$ for $t \in I$ then $\gamma'(t)$ forms the *tangent vector* at the point $\gamma(t)$ of γ.

- The tangent vector indicates the forward direction of γ relative to increasing parameter values.
- If γ is at unit speed then $\gamma'(t)$ forms a unit vector.
- A parametrization of the tangent line ℓ that passes through $\gamma(t)$ is given by

$$\ell(\lambda) = \gamma(t) + \lambda\gamma'(t) \quad \text{with } \lambda \in \mathbb{R}.$$

© M. Held (Univ. Salzburg)

Tangent Vector

Definition 68 (Tangent vector)

Let $\gamma\colon I \to \mathbb{R}^n$ be a C^1-curve. If $\gamma'(t) \neq 0$ for $t \in I$ then $\gamma'(t)$ forms the *tangent vector* at the point $\gamma(t)$ of γ.

- The tangent vector indicates the forward direction of γ relative to increasing parameter values.
- If γ is at unit speed then $\gamma'(t)$ forms a unit vector.
- A parametrization of the tangent line ℓ that passes through $\gamma(t)$ is given by

$$\ell(\lambda) = \gamma(t) + \lambda\gamma'(t) \quad \text{with } \lambda \in \mathbb{R}.$$

- If

$$\gamma(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

is a curve in $\mathbb{R}^2$ then the vector

$$\begin{pmatrix} -y'(t) \\ x'(t) \end{pmatrix}$$

is normal on the tangent line at $\gamma(t)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Curvature of Curves in $\mathbb{R}^3$

- The curvature at a given point of a curve (in $\mathbb{R}^3$) is a measure of how quickly the curve changes direction at that point relative to the speed of the curve.

Definition 69 (Curvature, Dt.: Krümmung)

Let $\gamma \colon I \to \mathbb{R}^3$ be a C^2 curve that is regular. The *curvature* $\kappa(t)$ of γ at the point $\gamma(t)$ is defined as

$$\kappa(t) := \frac{\|\gamma'(t) \times \gamma''(t)\|}{\|\gamma'(t)\|^3}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- The curvature at a given point of a curve (in $\mathbb{R}^3$) is a measure of how quickly the curve changes direction at that point relative to the speed of the curve.

Definition 69 (Curvature, Dt.: Krümmung)

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2 curve that is regular. The *curvature* $\kappa(t)$ of γ at the point $\gamma(t)$ is defined as

$$\kappa(t) := \frac{\|\gamma'(t) \times \gamma''(t)\|}{\|\gamma'(t)\|^3}.$$

Definition 70 (Radius of curvature, Dt.: Krümmungsradius)

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2 curve that is regular. If $\kappa(t) > 0$ then the *radius of curvature* $\rho(t)$ at the point $\gamma(t)$ is defined as

$$\rho(t) := \frac{1}{\kappa(t)}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 71 (Point of inflection, Dt.: Wendepunkt)

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2-curve that is regular. If for all $t \in I$ the second derivative γ'' does not vanish, i.e., if $\gamma''(t) \neq 0$, then a point $\gamma(t)$ for which $\kappa(t) = 0$ holds is called a *point of inflection* of γ.

Definition 71 (Point of inflection, Dt.: Wendepunkt)

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2-curve that is regular. If for all $t \in I$ the second derivative γ'' does not vanish, i.e., if $\gamma''(t) \neq 0$, then a point $\gamma(t)$ for which $\kappa(t) = 0$ holds is called a *point of inflection* of γ.

Lemma 72

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2-curve that is regular such that for all $t \in I$ the second derivative γ'' does not vanish. Then $\gamma(t)$ is a point of inflection of γ if and only if $\gamma'(t)$ and $\gamma''(t)$ are collinear.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 71 (Point of inflection, Dt.: Wendepunkt)

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2-curve that is regular. If for all $t \in I$ the second derivative γ'' does not vanish, i.e., if $\gamma''(t) \neq 0$, then a point $\gamma(t)$ for which $\kappa(t) = 0$ holds is called a *point of inflection* of γ.

Lemma 72

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2-curve that is regular such that for all $t \in I$ the second derivative γ'' does not vanish. Then $\gamma(t)$ is a point of inflection of γ if and only if $\gamma'(t)$ and $\gamma''(t)$ are collinear.

- Hence, at a point of inflection the radius of curvature is infinite and the circle of curvature degenerates to the tangent.

Lemma 73

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2-curve at unit speed that is regular. Then the following simplified formula holds:

$$\kappa(t) = \|\gamma''(t)\|$$

Curvature of Curves in $\mathbb{R}^3$

Lemma 73

Let $\gamma\colon I \to \mathbb{R}^3$ be a C^2-curve at unit speed that is regular. Then the following simplified formula holds:

$$\kappa(t) = \|\gamma''(t)\|$$

Sketch of proof : Recall that, in general,

$$\kappa(t) = \frac{\|\gamma'(t) \times \gamma''(t)\|}{\|\gamma'(t)\|^3}.$$

□

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 74

Let $\gamma\colon I \to \mathbb{R}^2$ be a C^2-curve that is regular, with $\gamma(t) = (x(t), y(t))$. Then $\kappa(t)$ of γ at the point $\gamma(t)$ is given as

$$\kappa(t) = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{((x'(t))^2 + (y'(t))^2)^{3/2}}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 74

Let $\gamma\colon I \to \mathbb{R}^2$ be a C^2-curve that is regular, with $\gamma(t) = (x(t), y(t))$. Then $\kappa(t)$ of γ at the point $\gamma(t)$ is given as

$$\kappa(t) = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{((x'(t))^2 + (y'(t))^2)^{3/2}}.$$

Sketch of proof : Recall that, in general,

$$\kappa(t) = \frac{\|\gamma'(t) \times \gamma''(t)\|}{\|\gamma'(t)\|^3}.$$

□

Curvature of Curves in $\mathbb{R}^2$

Lemma 74

Let $\gamma\colon I \to \mathbb{R}^2$ be a C^2-curve that is regular, with $\gamma(t) = (x(t), y(t))$. Then $\kappa(t)$ of γ at the point $\gamma(t)$ is given as

$$\kappa(t) = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{((x'(t))^2 + (y'(t))^2)^{3/2}}.$$

Sketch of proof: Recall that, in general,

$$\kappa(t) = \frac{\|\gamma'(t) \times \gamma''(t)\|}{\|\gamma'(t)\|^3}.$$

□

Corollary 75

Let $\gamma\colon I \to \mathbb{R}^2$ be a C^2-curve that is regular, with $\gamma(t) = (t, y(t))$. Then $\kappa(t)$ of γ at the point $\gamma(t)$ is given as

$$\kappa(t) = \frac{|y''(t)|}{(1 + (y'(t))^2)^{3/2}}.$$

Implications of Convexity in the Plane

Lemma 76

Every convex curve is simple.

Implications of Convexity in the Plane

Lemma 76

Every convex curve is simple.

Lemma 77

A convex Jordan curve bounds a convex area.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Implications of Convexity in the Plane

Lemma 76

Every convex curve is simple.

Lemma 77

A convex Jordan curve bounds a convex area.

Lemma 78

A smooth Jordan curve is convex if and only if its curvature has a consistent sign.

Implications of Convexity in the Plane

Lemma 76

Every convex curve is simple.

Lemma 77

A convex Jordan curve bounds a convex area.

Lemma 78

A smooth Jordan curve is convex if and only if its curvature has a consistent sign.

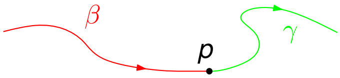
Lemma 79

Every bounded convex curve is rectifiable.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

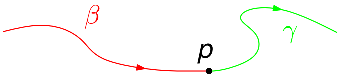
Parametric Continuity of a Curve

- Consider two curves $\beta\colon [a, b] \to \mathbb{R}^n$ and $\gamma\colon [c, d] \to \mathbb{R}^n$.
- Suppose that $\beta(b) = \gamma(c) =: p$.
- We are interested in checking how "smoothly" β and γ join at the joint p.



Parametric Continuity of a Curve

- Consider two curves $\beta\colon [a,b] \to \mathbb{R}^n$ and $\gamma\colon [c,d] \to \mathbb{R}^n$.
- Suppose that $\beta(b) = \gamma(c) =: p$.
- We are interested in checking how “smoothly” β and γ join at the joint p.



Definition 80 (C^k-continuous at joint, Dt.: C^k-stetiger Übergang)

Let $\beta\colon [a,b] \to \mathbb{R}^n$ and $\gamma\colon [c,d] \to \mathbb{R}^n$ be C^k-curves. If

$$\beta^{(i)}(b) = \gamma^{(i)}(c) \qquad \text{for all } i \in \{0,\ldots,k\}$$

then β and γ are C^k*-continuous at joint* $p := \beta(b)$.

- Of course, one-sided derivatives are to be considered in Def. 80.

Parametric Continuity of a Curve

- C^0-continuity implies that the end point of one curve is the start point of the second curve, i.e., they share a common *joint*.
- C^1-continuity implies that the speed does not change at p.
- C^2-continuity implies that the acceleration does not change at p.

Parametric Continuity of a Curve

- C^0-continuity implies that the end point of one curve is the start point of the second curve, i.e., they share a common *joint*.
- C^1-continuity implies that the speed does not change at p.
- C^2-continuity implies that the acceleration does not change at p.

Definition 81 (Curvature continuous, Dt.: krümmungsstetig)

Let $\beta\colon [a, b] \to \mathbb{R}^3$ and $\gamma\colon [c, d] \to \mathbb{R}^3$ be C^2-curves, with $\beta(b) = \gamma(c) =: p$. If the curvatures of β and γ are equal at p then β and γ are said to be *curvature continuous* at p.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Parametric Continuity of a Curve

- C^0-continuity implies that the end point of one curve is the start point of the second curve, i.e., they share a common *joint*.
- C^1-continuity implies that the speed does not change at p.
- C^2-continuity implies that the acceleration does not change at p.

Definition 81 (Curvature continuous, Dt.: krümmungsstetig)

Let $\beta\colon [a, b] \to \mathbb{R}^3$ and $\gamma\colon [c, d] \to \mathbb{R}^3$ be C^2-curves, with $\beta(b) = \gamma(c) =: p$. If the curvatures of β and γ are equal at p then β and γ are said to be *curvature continuous* at p.

Caveat

C^1-continuity plus curvature continuity need not imply C^2-continuity!

- Unfortunately, this important fact is missed frequently, and curvature continuity is often (wrongly) taken as a synonym for C^2-continuity ...

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Problems with Parametric Continuity

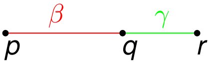
- Parametric continuity depends on the particular parametrizations of β and γ.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Problems with Parametric Continuity

- Parametric continuity depends on the particular parametrizations of β and γ.
- Consider three collinear points p, q, and r which define two straight-line segments joined at their common endpoint q:

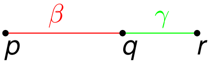
$$\beta(t) := p + t(q - p), \qquad t \in [0, 1]$$
$$\gamma(t) := q + (t - 1)(r - q), \qquad t \in [1, 2]$$



Problems with Parametric Continuity

- Parametric continuity depends on the particular parametrizations of β and γ.
- Consider three collinear points p, q, and r which define two straight-line segments joined at their common endpoint q:

$$\beta(t) := p + t(q - p), \qquad t \in [0, 1]$$
$$\gamma(t) := q + (t - 1)(r - q), \qquad t \in [1, 2]$$



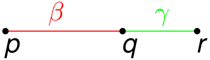
- Of course, β and γ are C^0-continuous at q.
- However, $\beta'(1) = q - p$ while $\gamma'(1) = r - q$. Thus, in general, β and γ will not be C^1-continuous at q.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Problems with Parametric Continuity

- Parametric continuity depends on the particular parametrizations of β and γ.
- Consider three collinear points p, q, and r which define two straight-line segments joined at their common endpoint q:

$$\beta(t) := p + t(q - p), \qquad t \in [0, 1]$$
$$\gamma(t) := q + (t - 1)(r - q), \qquad t \in [1, 2]$$



- Of course, β and γ are C^0-continuous at q.
- However, $\beta'(1) = q - p$ while $\gamma'(1) = r - q$. Thus, in general, β and γ will not be C^1-continuous at q.
- C^1-continuity at q could be achieved by resorting to arc-length parametrizations for β and γ:

$$\beta(t) := p + \frac{t}{\|q - p\|}(q - p), \qquad t \in [0, \|q - p\|]$$
$$\gamma(t) := q + \frac{t - \|q - p\|}{\|r - q\|}(r - q), \qquad t \in [\|q - p\|, \|q - p\| + \|r - q\|]$$

Geometric Continuity

- G^0-continuity equals C^0-continuity: The curves β and γ share a common *joint* p.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Geometric Continuity

- G^0-continuity equals C^0-continuity: The curves β and γ share a common *joint* p.

Definition 82 (G^1-continuous at joint, Dt.: G^1-stetiger Übergang)

Let $\beta\colon [a,b] \to \mathbb{R}^n$ and $\gamma\colon [c,d] \to \mathbb{R}^n$ be C^1-curves, with $\beta(b) = \gamma(c) =: p$. If

$$0 \neq \beta'(b) = \lambda \cdot \gamma'(c) \qquad \text{for some } \lambda \in \mathbb{R}^+$$

then β and γ are G^1*-continuous at joint* p.

- G^1-continuity means that β and γ share the tangent line at p.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Geometric Continuity

- G^0-continuity equals C^0-continuity: The curves β and γ share a common *joint* p.

Definition 82 (G^1-continuous at joint, Dt.: G^1-stetiger Übergang)

Let $\beta\colon [a,b] \to \mathbb{R}^n$ and $\gamma\colon [c,d] \to \mathbb{R}^n$ be C^1-curves, with $\beta(b) = \gamma(c) =: p$. If

$$0 \neq \beta'(b) = \lambda \cdot \gamma'(c) \qquad \text{for some } \lambda \in \mathbb{R}^+$$

then β and γ are *G^1-continuous at joint p*.

- G^1-continuity means that β and γ share the tangent line at p.
- Higher-order geometric continuities are a bit tricky to define formally for $k \geq 2$.
- G^2-continuity means that β and γ share the tangent line and also the curvature at p.

- G^0-continuity equals C^0-continuity: The curves β and γ share a common *joint* p.

Definition 82 (G^1-continuous at joint, Dt.: G^1-stetiger Übergang)

Let $\beta\colon [a,b] \to \mathbb{R}^n$ and $\gamma\colon [c,d] \to \mathbb{R}^n$ be C^1-curves, with $\beta(b) = \gamma(c) =: p$. If

$$0 \neq \beta'(b) = \lambda \cdot \gamma'(c) \qquad \text{for some } \lambda \in \mathbb{R}^+$$

then β and γ are G^1*-continuous at joint* p.

- G^1-continuity means that β and γ share the tangent line at p.
- Higher-order geometric continuities are a bit tricky to define formally for $k \geq 2$.
- G^2-continuity means that β and γ share the tangent line and also the curvature at p.
- In general, G^k-continuity exists at p if β and γ can be reparameterized such that they join with C^k-continuity at p.
- C^k-continuity implies G^k-continuity.

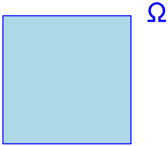
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

2 Mathematics for Geometric Modeling

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

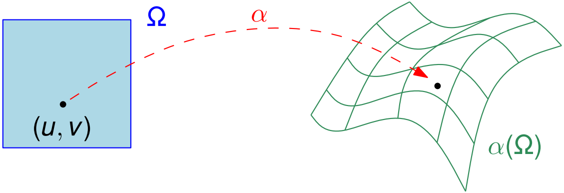
Definition 83 (Parametric surface)

Let $\Omega \subseteq \mathbb{R}^2$.



Definition 83 (Parametric surface)

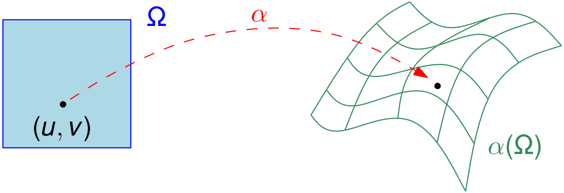
Let $\Omega \subseteq \mathbb{R}^2$. A continuous mapping $\alpha \colon \Omega \to \mathbb{R}^3$ is called a *parametrization* of $\alpha(\Omega)$, and $\alpha(\Omega)$ is called the (parametric) *surface* parameterized by α.



Definition 83 (Parametric surface)

Let $\Omega \subseteq \mathbb{R}^2$. A continuous mapping $\alpha\colon \Omega \to \mathbb{R}^3$ is called a *parametrization* of $\alpha(\Omega)$, and $\alpha(\Omega)$ is called the (parametric) *surface* parameterized by α.

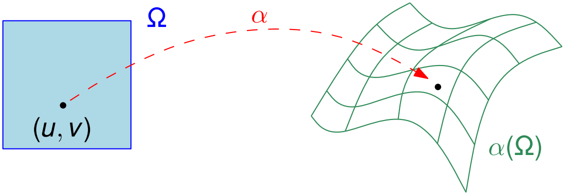
- For instance, every point on the surface of Earth can be described by the geographic coordinates longitude and latitude.



Definition 83 (Parametric surface)

Let $\Omega \subseteq \mathbb{R}^2$. A continuous mapping $\alpha\colon \Omega \to \mathbb{R}^3$ is called a *parametrization* of $\alpha(\Omega)$, and $\alpha(\Omega)$ is called the (parametric) *surface* parameterized by α.

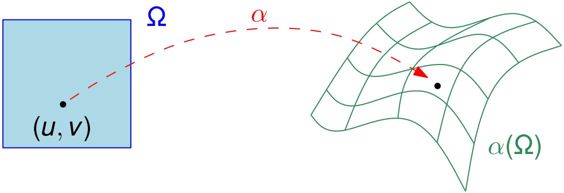
- For instance, every point on the surface of Earth can be described by the geographic coordinates longitude and latitude.
- Parametrizations of a surface (regarded as a set $\mathcal{S} \subset \mathbb{R}^3$) need not be unique: two different parametrizations α and β may exist such that $\mathcal{S} = \alpha(\Omega_1) = \beta(\Omega_2)$.



Definition 83 (Parametric surface)

Let $\Omega \subseteq \mathbb{R}^2$. A continuous mapping $\alpha\colon \Omega \to \mathbb{R}^3$ is called a *parametrization* of $\alpha(\Omega)$, and $\alpha(\Omega)$ is called the (parametric) *surface* parameterized by α.

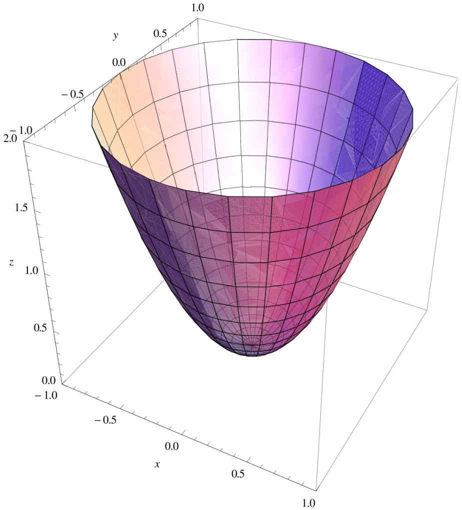
- For instance, every point on the surface of Earth can be described by the geographic coordinates longitude and latitude.
- Parametrizations of a surface (regarded as a set $\mathcal{S} \subset \mathbb{R}^3$) need not be unique: two different parametrizations α and β may exist such that $\mathcal{S} = \alpha(\Omega_1) = \beta(\Omega_2)$.
- For simplicity, we will not distinguish between a surface and one of its parametrizations if the meaning is clear.



Sample Parametric Surface: Frustum of a Paraboloid

$$\alpha\colon [0,1]\times[0,2\pi]\to\mathbb{R}^3$$

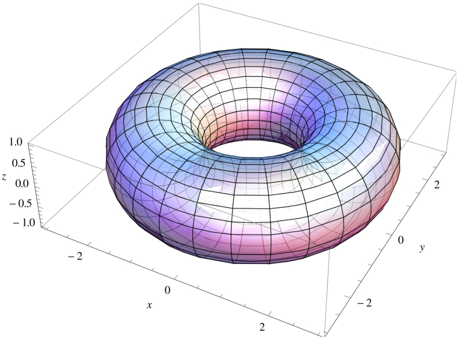
$$\alpha(u,v) := \begin{pmatrix} u\cos v \\ u\sin v \\ 2u^2 \end{pmatrix}$$



Sample Parametric Surface: Torus

$$\alpha\colon [0, 2\pi]^2 \to \mathbb{R}^3$$

$$\alpha(u, v) := \begin{pmatrix} (2 + \cos v) \cos u \\ (2 + \cos v) \sin u \\ \sin v \end{pmatrix}$$



Definition 84 (Regular parametrization, Dt.: reguläre (od. zulässige) Param.)

Let $\Omega \subseteq \mathbb{R}^2$. A continuous mapping $\alpha\colon \Omega \to \mathbb{R}^3$ in the variables u and v is called a *regular parametrization* of $\alpha(\Omega)$ if

1. α is (continuously) differentiable on Ω,
2. $\frac{\partial\alpha}{\partial u}(u_0, v_0)$ and $\frac{\partial\alpha}{\partial v}(u_0, v_0)$ are linearly independent for all (u_0, v_0) in Ω.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 84 (Regular parametrization, Dt.: reguläre (od. zulässige) Param.)

Let $\Omega \subseteq \mathbb{R}^2$. A continuous mapping $\alpha\colon \Omega \to \mathbb{R}^3$ in the variables u and v is called a *regular parametrization* of $\alpha(\Omega)$ if

1. α is (continuously) differentiable on Ω,
2. $\frac{\partial \alpha}{\partial u}(u_0, v_0)$ and $\frac{\partial \alpha}{\partial v}(u_0, v_0)$ are linearly independent for all (u_0, v_0) in Ω.

- Note that $\frac{\partial \alpha}{\partial u}(u_0, v_0)$ and $\frac{\partial \alpha}{\partial v}(u_0, v_0)$ are linearly independent if and only if

$$\frac{\partial \alpha}{\partial u}(u_0, v_0) \times \frac{\partial \alpha}{\partial v}(u_0, v_0) \neq 0.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Basic Definitions for Parametric Surfaces

Definition 84 (Regular parametrization, Dt.: reguläre (od. zulässige) Param.)

Let $\Omega \subseteq \mathbb{R}^2$. A continuous mapping $\alpha\colon \Omega \to \mathbb{R}^3$ in the variables u and v is called a *regular parametrization* of $\alpha(\Omega)$ if

1. α is (continuously) differentiable on Ω,
2. $\frac{\partial \alpha}{\partial u}(u_0, v_0)$ and $\frac{\partial \alpha}{\partial v}(u_0, v_0)$ are linearly independent for all (u_0, v_0) in Ω.

- Note that $\frac{\partial \alpha}{\partial u}(u_0, v_0)$ and $\frac{\partial \alpha}{\partial v}(u_0, v_0)$ are linearly independent if and only if

$$\frac{\partial \alpha}{\partial u}(u_0, v_0) \times \frac{\partial \alpha}{\partial v}(u_0, v_0) \neq 0.$$

Definition 85 (Singular point, Dt.: singulärer Punkt)

Let $\Omega \subseteq \mathbb{R}^2$. A point $(u_0, v_0) \in \Omega$ is a *singular point* of a (continuously) differentiable parametrization $\alpha\colon \Omega \to \mathbb{R}^3$ if $\frac{\partial \alpha}{\partial u}(u_0, v_0)$ and $\frac{\partial \alpha}{\partial v}(u_0, v_0)$ are linearly dependent.

Tangent Plane and Normal Vector

Definition 86 (Tangent plane, Dt.: Tangentialebene)

Consider a regular parametrization $\alpha\colon \Omega \to \mathbb{R}^3$ of a surface $\mathcal{S}$. For $(u, v) \in \Omega$, the *tangent plane* $\varepsilon(u, v)$ of $\mathcal{S}$ at $\alpha(u, v)$ is the plane through $\alpha(u, v)$ that is spanned by the vectors

$$\frac{\partial\alpha}{\partial u}(u, v) \qquad \text{and} \qquad \frac{\partial\alpha}{\partial v}(u, v).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 86 (Tangent plane, Dt.: Tangentialebene)

Consider a regular parametrization $\alpha\colon \Omega \to \mathbb{R}^3$ of a surface $\mathcal{S}$. For $(u, v) \in \Omega$, the *tangent plane* $\varepsilon(u, v)$ of $\mathcal{S}$ at $\alpha(u, v)$ is the plane through $\alpha(u, v)$ that is spanned by the vectors

$$\frac{\partial\alpha}{\partial u}(u, v) \qquad \text{and} \qquad \frac{\partial\alpha}{\partial v}(u, v).$$

Definition 87 (Normal vector, Dt.: Normalvektor)

Consider a regular parametrization $\alpha\colon \Omega \to \mathbb{R}^3$ of a surface $\mathcal{S}$. For $(u, v) \in \Omega$, the *normal vector* $N(u, v)$ of $\mathcal{S}$ at $\alpha(u, v)$ is given by

$$N(u, v) := \frac{\partial\alpha}{\partial u}(u, v) \times \frac{\partial\alpha}{\partial v}(u, v).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Curves on Surfaces

- Suppose that $\Omega = [u_{min}, u_{max}] \times [v_{min}, v_{max}]$
- If we fix $v := v_0 \in [v_{min}, v_{max}]$ and let u vary, then $\alpha(u, v_0)$ depends on one parameter; it is called an *isoparametric curve* or, more specifically, the *u-parameter curve*.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Curves on Surfaces

- Suppose that $\Omega = [u_{min}, u_{max}] \times [v_{min}, v_{max}]$
- If we fix $v := v_0 \in [v_{min}, v_{max}]$ and let u vary, then $\alpha(u, v_0)$ depends on one parameter; it is called an *isoparametric curve* or, more specifically, the *u-parameter curve*.
- Likewise, we can fix $u := u_0 \in [u_{min}, u_{max}]$ and let v vary to obtain the *v-parameter curve* $\alpha(u_0, v)$.

Curves on Surfaces

- Suppose that $\Omega = [u_{min}, u_{max}] \times [v_{min}, v_{max}]$
- If we fix $v := v_0 \in [v_{min}, v_{max}]$ and let u vary, then $\alpha(u, v_0)$ depends on one parameter; it is called an *isoparametric curve* or, more specifically, the *u-parameter curve*.
- Likewise, we can fix $u := u_0 \in [u_{min}, u_{max}]$ and let v vary to obtain the *v-parameter curve* $\alpha(u_0, v)$.
- Tangent vectors for the u-parameter and v-parameter curves are computed by partial derivatives of α with respect to u and v, respectively:

$$\frac{\partial \alpha}{\partial u}(u, v) \qquad \text{for } v := v_0$$

$$\frac{\partial \alpha}{\partial v}(u, v) \qquad \text{for } u := u_0$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Curves on Surfaces

- The standard parametrization of the unit sphere is given by

$$\alpha(u, v) := \begin{pmatrix} \cos u \cdot \cos v \\ \sin u \cdot \cos v \\ \sin v \end{pmatrix} \qquad \text{with } (u, v) \in [0, 2\pi[\times[-\frac{\pi}{2}, \frac{\pi}{2}].$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- The standard parametrization of the unit sphere is given by

$$\alpha(u, v) := \begin{pmatrix} \cos u \cdot \cos v \\ \sin u \cdot \cos v \\ \sin v \end{pmatrix} \qquad \text{with } (u, v) \in [0, 2\pi[\times[-\frac{\pi}{2}, \frac{\pi}{2}].$$

- We get

$$\frac{\partial \alpha}{\partial u}(u, v_0) = \begin{pmatrix} -\sin u \cdot \cos v_0 \\ \cos u \cdot \cos v_0 \\ 0 \end{pmatrix}$$

Curves on Surfaces

- The standard parametrization of the unit sphere is given by

$$\alpha(u, v) := \begin{pmatrix} \cos u \cdot \cos v \\ \sin u \cdot \cos v \\ \sin v \end{pmatrix} \qquad \text{with } (u, v) \in [0, 2\pi[\times[-\frac{\pi}{2}, \frac{\pi}{2}].$$

- We get

$$\frac{\partial\alpha}{\partial u}(u, v_0) = \begin{pmatrix} -\sin u \cdot \cos v_0 \\ \cos u \cdot \cos v_0 \\ 0 \end{pmatrix}$$

and

$$\frac{\partial\alpha}{\partial v}(u_0, v) = \begin{pmatrix} -\cos u_0 \cdot \sin v \\ -\sin u_0 \cdot \sin v \\ \cos v \end{pmatrix}.$$

Curves on Surfaces

- The standard parametrization of the unit sphere is given by

$$\alpha(u,v) := \begin{pmatrix} \cos u \cdot \cos v \\ \sin u \cdot \cos v \\ \sin v \end{pmatrix} \qquad \text{with } (u,v) \in [0, 2\pi[\times[-\frac{\pi}{2}, \frac{\pi}{2}].$$

- We get

$$\frac{\partial \alpha}{\partial u}(u, v_0) = \begin{pmatrix} -\sin u \cdot \cos v_0 \\ \cos u \cdot \cos v_0 \\ 0 \end{pmatrix}$$

and

$$\frac{\partial \alpha}{\partial v}(u_0, v) = \begin{pmatrix} -\cos u_0 \cdot \sin v \\ -\sin u_0 \cdot \sin v \\ \cos v \end{pmatrix}.$$

- Note that $\frac{\partial \alpha}{\partial u}(u_0, v_0) \perp \frac{\partial \alpha}{\partial v}(u_0, v_0)$.

- The standard parametrization of the unit sphere is given by

$$\alpha(u,v) := \begin{pmatrix} \cos u \cdot \cos v \\ \sin u \cdot \cos v \\ \sin v \end{pmatrix} \qquad \text{with } (u,v) \in [0, 2\pi[\times[-\frac{\pi}{2}, \frac{\pi}{2}].$$

- We get

$$\frac{\partial\alpha}{\partial u}(u, v_0) = \begin{pmatrix} -\sin u \cdot \cos v_0 \\ \cos u \cdot \cos v_0 \\ 0 \end{pmatrix}$$

and

$$\frac{\partial\alpha}{\partial v}(u_0, v) = \begin{pmatrix} -\cos u_0 \cdot \sin v \\ -\sin u_0 \cdot \sin v \\ \cos v \end{pmatrix}.$$

- Note that $\frac{\partial\alpha}{\partial u}(u_0, v_0) \perp \frac{\partial\alpha}{\partial v}(u_0, v_0)$.
- Also, note that $\frac{\partial\alpha}{\partial u}(u, v_0)$ vanishes for $v_0 := \pm\frac{\pi}{2}$. Hence, the north and south poles are singular points of this parametrization.

Practical Continuity Requirements

- Parametric continuity of curves is important for animations: If an object moves along curve β with constant speed, then there should be no sudden increase in speed once it moves along γ. Thus, C^1 continuity is required.

Practical Continuity Requirements

- Parametric continuity of curves is important for animations: If an object moves along curve β with constant speed, then there should be no sudden increase in speed once it moves along γ. Thus, C^1 continuity is required.
- Roads and railroad tracks have so-called transition curves (such as clothoids) to lead from a straight segment to a circular segments, or to connect arcs of different radii, thus achieving (at least) G^2 continuity.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

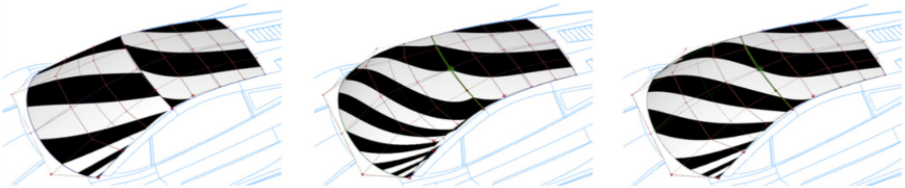
Practical Continuity Requirements

- Parametric continuity of curves is important for animations: If an object moves along curve β with constant speed, then there should be no sudden increase in speed once it moves along γ. Thus, C^1 continuity is required.
- Roads and railroad tracks have so-called transition curves (such as clothoids) to lead from a straight segment to a circular segments, or to connect arcs of different radii, thus achieving (at least) G^2 continuity.
- The definitions of C^k continuity and G^k continuity can be extended to surface patches.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Practical Continuity Requirements

- Parametric continuity of curves is important for animations: If an object moves along curve β with constant speed, then there should be no sudden increase in speed once it moves along γ. Thus, C^1 continuity is required.
- Roads and railroad tracks have so-called transition curves (such as clothoids) to lead from a straight segment to a circular segments, or to connect arcs of different radii, thus achieving (at least) G^2 continuity.
- The definitions of C^k continuity and G^k continuity can be extended to surface patches.
- Reflections on a surface (e.g., a car body) will not appear smooth unless G^2-continuity is achieved between neighboring patches: “Class-A surface”.



[Image credit: © Autodesk]

3 Bézier Curves and Surfaces

- Bernstein Basis Polynomials
- Bézier Curves
- Bézier Surfaces

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

3 **Bézier Curves and Surfaces**

- Bernstein Basis Polynomials
  - Definition and Properties
  - Basis Conversions
- Bézier Curves
- Bézier Surfaces

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 88 (Bernstein basis polynomials)

The $n+1$ *Bernstein basis polynomials* of degree n, for $n \in \mathbb{N}_0$, are defined as

$$B_{k,n}(x) := \binom{n}{k} x^k (1-x)^{n-k} \quad \text{for } k \in \{0, 1, \ldots, n\}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 88 (Bernstein basis polynomials)

The $n+1$ *Bernstein basis polynomials* of degree n, for $n \in \mathbb{N}_0$, are defined as

$$B_{k,n}(x) := \binom{n}{k} x^k (1-x)^{n-k} \quad \text{for } k \in \{0, 1, \ldots, n\}.$$

- We use the convention $0^0 := 1$.
- For convenience purposes, we define $B_{k,n}(x) := 0$ for $k < 0$ or $k > n$.

Definition 88 (Bernstein basis polynomials)

The $n+1$ *Bernstein basis polynomials* of degree n, for $n \in \mathbb{N}_0$, are defined as

$$B_{k,n}(x) := \binom{n}{k} x^k (1-x)^{n-k} \quad \text{for } k \in \{0, 1, \ldots, n\}.$$

- We use the convention $0^0 := 1$.
- For convenience purposes, we define $B_{k,n}(x) := 0$ for $k < 0$ or $k > n$.
- $B_{0,0}(x) = 1$.
- $B_{0,1}(x) = 1 - x$ and $B_{1,1}(x) = x$.
- $B_{0,2}(x) = (1-x)^2$ and $B_{1,2}(x) = 2x(1-x)$ and $B_{2,2}(x) = x^2$.

Bernstein Basis Polynomials

Definition 88 (Bernstein basis polynomials)

The $n+1$ *Bernstein basis polynomials* of degree n, for $n \in \mathbb{N}_0$, are defined as

$$B_{k,n}(x) := \binom{n}{k} x^k (1-x)^{n-k} \quad \text{for } k \in \{0, 1, \ldots, n\}.$$

- We use the convention $0^0 := 1$.
- For convenience purposes, we define $B_{k,n}(x) := 0$ for $k < 0$ or $k > n$.
- $B_{0,0}(x) = 1$.
- $B_{0,1}(x) = 1 - x$ and $B_{1,1}(x) = x$.
- $B_{0,2}(x) = (1-x)^2$ and $B_{1,2}(x) = 2x(1-x)$ and $B_{2,2}(x) = x^2$.

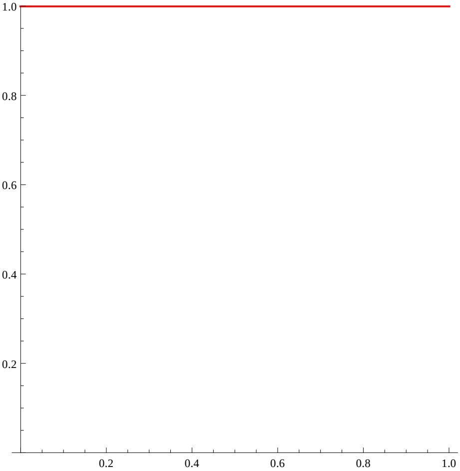
- Introduced by Sergei N. Bernstein in 1911 for a constructive proof of Weierstrass' Approximation Theorem 198.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials

- All Bernstein basis polynomials of degree $n = 0$ over the interval $[0, 1]$:

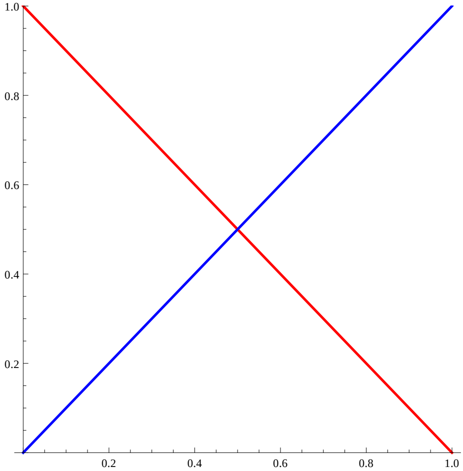
  1



Bernstein Basis Polynomials

- All Bernstein basis polynomials of degree $n = 1$ over the interval $[0, 1]$:

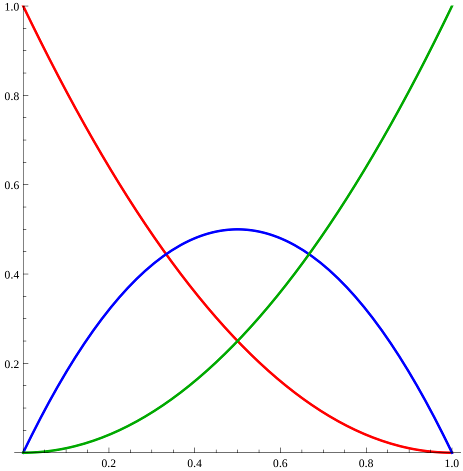
$1 - x$ x



Bernstein Basis Polynomials

- All Bernstein basis polynomials of degree $n = 2$ over the interval $[0, 1]$:

$(1 - x)^2$ $2x(1 - x)$ x^2

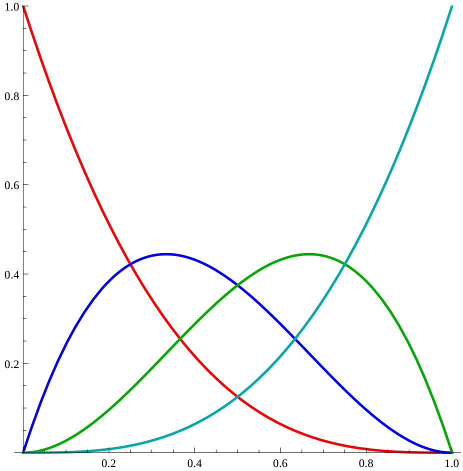


UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials

- All Bernstein basis polynomials of degree $n = 3$ over the interval $[0, 1]$:

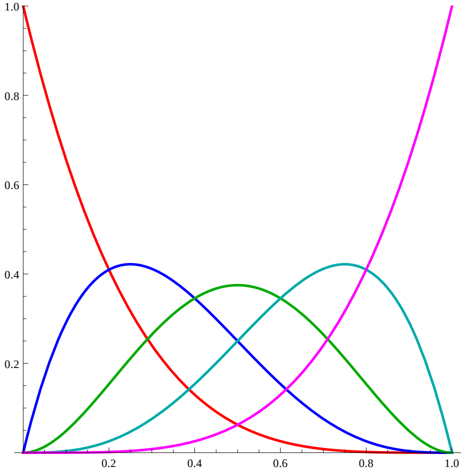
$(1-x)^3$ $\quad 3x(1-x)^2$ $\quad 3x^2(1-x)$ $\quad x^3$



Bernstein Basis Polynomials

- All Bernstein basis polynomials of degree $n = 4$ over the interval $[0, 1]$:

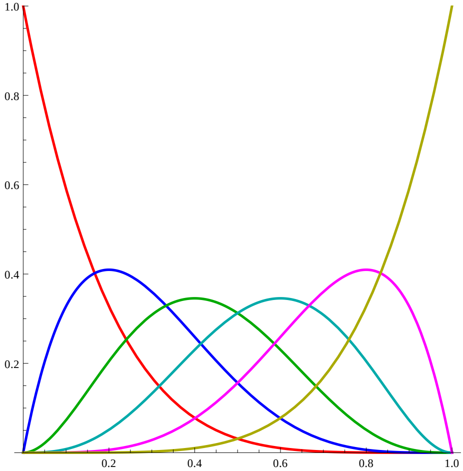
$(1-x)^4$ $\quad 4x(1-x)^3$ $\quad 6x^2(1-x)^2$ $\quad 4x^3(1-x)$ $\quad x^4$



Bernstein Basis Polynomials

- All Bernstein basis polynomials of degree $n = 5$ over the interval $[0, 1]$:

  $(1-x)^5$ $5x(1-x)^4$ $10x^2(1-x)^3$ $10x^3(1-x)^2$ $5x^4(1-x)$ x^5



Recursion Formula for Bernstein Basis Polynomials

Lemma 89

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n}(x)$ of degree n can be written as the sum of two basis polynomials of degree $n-1$:

$$B_{k,n}(x) = x \cdot B_{k-1,n-1}(x) + (1-x) \cdot B_{k,n-1}(x)$$

Recursion Formula for Bernstein Basis Polynomials

Lemma 89

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n}(x)$ of degree n can be written as the sum of two basis polynomials of degree $n-1$:

$$B_{k,n}(x) = x \cdot B_{k-1,n-1}(x) + (1-x) \cdot B_{k,n-1}(x)$$

Proof: Let $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$ be arbitrary but fixed, and recall that

$$B_{k,n}(x) \stackrel{\text{Def. 88}}{=} \binom{n}{k} x^k (1-x)^{n-k} \qquad \text{and} \qquad \binom{n}{k} \stackrel{\text{Thm. 9}}{=} \binom{n-1}{k-1} + \binom{n-1}{k}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 89

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n}(x)$ of degree n can be written as the sum of two basis polynomials of degree $n-1$:

$$B_{k,n}(x) = x \cdot B_{k-1,n-1}(x) + (1-x) \cdot B_{k,n-1}(x)$$

Proof: Let $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$ be arbitrary but fixed, and recall that

$$B_{k,n}(x) \overset{\text{Def. 88}}{=} \binom{n}{k} x^k (1-x)^{n-k} \quad \text{and} \quad \binom{n}{k} \overset{\text{Thm. 9}}{=} \binom{n-1}{k-1} + \binom{n-1}{k}.$$

We get

$$B_{k,n}(x) = \binom{n-1}{k-1} x^k (1-x)^{n-k} + \binom{n-1}{k} x^k (1-x)^{n-k}$$

Recursion Formula for Bernstein Basis Polynomials

Lemma 89

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n}(x)$ of degree n can be written as the sum of two basis polynomials of degree $n-1$:

$$B_{k,n}(x) = x \cdot B_{k-1,n-1}(x) + (1-x) \cdot B_{k,n-1}(x)$$

Proof: Let $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$ be arbitrary but fixed, and recall that

$$B_{k,n}(x) \overset{\text{Def. 88}}{=} \binom{n}{k} x^k (1-x)^{n-k} \quad \text{and} \quad \binom{n}{k} \overset{\text{Thm. 9}}{=} \binom{n-1}{k-1} + \binom{n-1}{k}.$$

We get

$$\begin{aligned} B_{k,n}(x) &= \binom{n-1}{k-1} x^k (1-x)^{n-k} + \binom{n-1}{k} x^k (1-x)^{n-k} \\ &= x \binom{n-1}{k-1} x^{k-1} (1-x)^{(n-1)-(k-1)} + (1-x) \binom{n-1}{k} x^k (1-x)^{(n-1)-k} \end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 89

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n}(x)$ of degree n can be written as the sum of two basis polynomials of degree $n-1$:

$$B_{k,n}(x) = x \cdot B_{k-1,n-1}(x) + (1-x) \cdot B_{k,n-1}(x)$$

Proof : Let $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$ be arbitrary but fixed, and recall that

$$B_{k,n}(x) \overset{\text{Def. 88}}{=} \binom{n}{k} x^k (1-x)^{n-k} \qquad \text{and} \qquad \binom{n}{k} \overset{\text{Thm. 9}}{=} \binom{n-1}{k-1} + \binom{n-1}{k}.$$

We get

$$\begin{aligned} B_{k,n}(x) &= \binom{n-1}{k-1} x^k (1-x)^{n-k} + \binom{n-1}{k} x^k (1-x)^{n-k} \\ &= x \binom{n-1}{k-1} x^{k-1} (1-x)^{(n-1)-(k-1)} + (1-x) \binom{n-1}{k} x^k (1-x)^{(n-1)-k} \\ &= x \cdot B_{k-1,n-1}(x) + (1-x) \cdot B_{k,n-1}(x). \end{aligned}$$

Lemma 90

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n-1}$ can be written as linear combination of Bernstein basis polynomials of degree n:

$$B_{k,n-1}(x) = \frac{n-k}{n} B_{k,n}(x) + \frac{k+1}{n} B_{k+1,n}(x).$$

Lemma 90

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n-1}$ can be written as linear combination of Bernstein basis polynomials of degree n:

$$B_{k,n-1}(x) = \frac{n-k}{n} B_{k,n}(x) + \frac{k+1}{n} B_{k+1,n}(x).$$

Lemma 91

For all $n, k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n}$ is non-negative over the unit interval:

$$B_{k,n}(x) \geq 0 \qquad \text{for all } x \in [0,1].$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 90

For all $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n-1}$ can be written as linear combination of Bernstein basis polynomials of degree n:

$$B_{k,n-1}(x) = \frac{n-k}{n} B_{k,n}(x) + \frac{k+1}{n} B_{k+1,n}(x).$$

Lemma 91

For all $n, k \in \mathbb{N}_0$ with $k \leq n$, the Bernstein basis polynomial $B_{k,n}$ is non-negative over the unit interval:

$$B_{k,n}(x) \geq 0 \qquad \text{for all } x \in [0,1].$$

Proof: Recall the definition of the Bernstein basis polynomials:

$$B_{k,n}(x) \overset{\text{Def. 88}}{=} \binom{n}{k} \underbrace{(x)}_{\geq 0}{}^{k} \underbrace{(1-x)}_{\geq 0}{}^{n-k} \geq 0 \qquad \text{for all } x \in [0,1].$$

□

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bernstein Basis Polynomials

Lemma 92 (Partition of unity, Dt.: Zerlegung der Eins)

For all $n \in \mathbb{N}_0$, the $n+1$ Bernstein basis polynomials of degree n form a partition of unity, i.e., they sum up to one:

$$\sum_{k=0}^{n} B_{k,n}(x) = 1 \qquad \text{for all } x \in [0,1].$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 92 (Partition of unity, Dt.: Zerlegung der Eins)

For all $n \in \mathbb{N}_0$, the $n+1$ Bernstein basis polynomials of degree n form a partition of unity, i.e., they sum up to one:

$$\sum_{k=0}^{n} B_{k,n}(x) = 1 \qquad \text{for all } x \in [0,1].$$

Proof: Trivial for $n := 0$. Now recall the Binomial Theorem 10, for $a, b \in \mathbb{R}$ and $n \in \mathbb{N}$:

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bernstein Basis Polynomials

Lemma 92 (Partition of unity, Dt.: Zerlegung der Eins)

For all $n \in \mathbb{N}_0$, the $n+1$ Bernstein basis polynomials of degree n form a partition of unity, i.e., they sum up to one:

$$\sum_{k=0}^{n} B_{k,n}(x) = 1 \qquad \text{for all } x \in [0,1].$$

Proof : Trivial for $n := 0$. Now recall the Binomial Theorem 10, for $a, b \in \mathbb{R}$ and $n \in \mathbb{N}$:

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$$

Then the claim is an immediate consequence by setting $a := x$ and $b := 1 - x$:

$$1 = (x + (1-x))^n = \sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{n-k} = \sum_{k=0}^{n} B_{k,n}(x).$$

□

Lemma 93

For all $n \in \mathbb{N}_0$ and any set of $n+1$ points in $\mathbb{R}^2$ with position vectors $p_0, p_1, p_2, \ldots, p_n$, the term

$$B_{0,n}(t)p_0 + B_{1,n}(t)p_1 + \cdots + B_{n,n}(t)p_n$$

forms a convex combination of these points for all $t \in [0, 1]$.

Lemma 93

For all $n \in \mathbb{N}_0$ and any set of $n+1$ points in $\mathbb{R}^2$ with position vectors $p_0, p_1, p_2, \ldots, p_n$, the term

$$B_{0,n}(t)p_0 + B_{1,n}(t)p_1 + \cdots + B_{n,n}(t)p_n$$

forms a convex combination of these points for all $t \in [0, 1]$.

Proof : This is an immediate consequence of Lem. 91 and Lem. 92. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 93

For all $n \in \mathbb{N}_0$ and any set of $n+1$ points in $\mathbb{R}^2$ with position vectors $p_0, p_1, p_2, \ldots, p_n$, the term

$$B_{0,n}(t)p_0 + B_{1,n}(t)p_1 + \cdots + B_{n,n}(t)p_n$$

forms a convex combination of these points for all $t \in [0, 1]$.

Proof : This is an immediate consequence of Lem. 91 and Lem. 92. □

Corollary 94 (Convex hull property)

For all $n \in \mathbb{N}_0$ and any set of $n+1$ points in $\mathbb{R}^2$ with position vectors $p_0, p_1, p_2, \ldots, p_n$, the point

$$B_{0,n}(t)p_0 + B_{1,n}(t)p_1 + \cdots + B_{n,n}(t)p_n$$

lies within $CH(\{p_0, p_1, p_2, \ldots, p_n\})$ for all $t \in [0, 1]$.

Lemma 93

For all $n \in \mathbb{N}_0$ and any set of $n+1$ points in $\mathbb{R}^2$ with position vectors $p_0, p_1, p_2, \ldots, p_n$, the term

$$B_{0,n}(t)p_0 + B_{1,n}(t)p_1 + \cdots + B_{n,n}(t)p_n$$

forms a convex combination of these points for all $t \in [0,1]$.

Proof: This is an immediate consequence of Lem. 91 and Lem. 92. □

Corollary 94 (Convex hull property)

For all $n \in \mathbb{N}_0$ and any set of $n+1$ points in $\mathbb{R}^2$ with position vectors $p_0, p_1, p_2, \ldots, p_n$, the point

$$B_{0,n}(t)p_0 + B_{1,n}(t)p_1 + \cdots + B_{n,n}(t)p_n$$

lies within $CH(\{p_0, p_1, p_2, \ldots, p_n\})$ for all $t \in [0,1]$.

Proof: Recall Def. 22: $CH(\{p_0, p_1, p_2, \ldots, p_n\})$ equals the set of all convex combinations of $p_0, p_1, p_2, \ldots, p_n$.

Derivatives of Bernstein Basis Polynomials

Lemma 95

For $n, k \in \mathbb{N}_0$ and $i \in \mathbb{N}$ with $i \leq n$, the i-th derivative of $B_{k,n}(x)$ can be written as a linear combination of Bernstein basis polynomials of degree $n - i$:

$$B^{(i)}_{k,n}(x) = \frac{n!}{(n-i)!} \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} B_{k-j,n-i}(x)$$

Derivatives of Bernstein Basis Polynomials

Lemma 95

For $n, k \in \mathbb{N}_0$ and $i \in \mathbb{N}$ with $i \leq n$, the i-th derivative of $B_{k,n}(x)$ can be written as a linear combination of Bernstein basis polynomials of degree $n-i$:

$$B_{k,n}^{(i)}(x) = \frac{n!}{(n-i)!} \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} B_{k-j,n-i}(x)$$

Corollary 96

For $n, k \in \mathbb{N}_0$, the first and second derivative of $B_{k,n}(x)$ are given as follows:

$$B'_{k,n}(x) = n\big(B_{k-1,n-1}(x) - B_{k,n-1}(x)\big)$$
$$B''_{k,n}(x) = n(n-1)\big(B_{k-2,n-2}(x) - 2B_{k-1,n-2}(x) + B_{k,n-2}(x)\big)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials Form a Basis

Lemma 97

The $n+1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Bernstein Basis Polynomials Form a Basis

Lemma 97

The $n + 1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Proof : We do a proof by induction.
I.B.: The claim is obviously true for $n := 0$ and $n := 1$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials Form a Basis

Lemma 97

The $n+1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Proof: We do a proof by induction.
I.B.: The claim is obviously true for $n := 0$ and $n := 1$.
I.H.: Suppose that the claim is true for an arbitrary but fixed $n-1 \in \mathbb{N}_0$, i.e., that $\sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) = 0$ implies $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = 0$.

Lemma 97

The $n+1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Proof : We do a proof by induction.
I.B.: The claim is obviously true for $n := 0$ and $n := 1$.
I.H.: Suppose that the claim is true for an arbitrary but fixed $n-1 \in \mathbb{N}_0$, i.e., that $\sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) = 0$ implies $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = 0$.
I.S.: Suppose that $\sum_{k=0}^{n} \lambda_k B_{k,n}(x) = 0$ for some $\lambda_0, \lambda_1, \ldots, \lambda_n \in \mathbb{R}$. Then we get

$$0 = \sum_{k=0}^{n} \lambda_k B'_{k,n}(x)$$

Bernstein Basis Polynomials Form a Basis

Lemma 97

The $n+1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Proof : We do a proof by induction.
I.B.: The claim is obviously true for $n := 0$ and $n := 1$.
I.H.: Suppose that the claim is true for an arbitrary but fixed $n-1 \in \mathbb{N}_0$, i.e., that $\sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) = 0$ implies $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = 0$.
I.S.: Suppose that $\sum_{k=0}^{n} \lambda_k B_{k,n}(x) = 0$ for some $\lambda_0, \lambda_1, \ldots, \lambda_n \in \mathbb{R}$. Then we get

$$0 = \sum_{k=0}^{n} \lambda_k B'_{k,n}(x) \overset{\text{Lem. } 95}{=} \sum_{k=0}^{n} \lambda_k \cdot n \cdot (B_{k-1,n-1}(x) - B_{k,n-1}(x))$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials Form a Basis

Lemma 97

The $n+1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Proof: We do a proof by induction.
I.B.: The claim is obviously true for $n := 0$ and $n := 1$.
I.H.: Suppose that the claim is true for an arbitrary but fixed $n-1 \in \mathbb{N}_0$, i.e., that $\sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) = 0$ implies $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = 0$.
I.S.: Suppose that $\sum_{k=0}^{n} \lambda_k B_{k,n}(x) = 0$ for some $\lambda_0, \lambda_1, \ldots, \lambda_n \in \mathbb{R}$. Then we get

$$0 = \sum_{k=0}^{n} \lambda_k B'_{k,n}(x) \overset{Lem.~95}{=} \sum_{k=0}^{n} \lambda_k \cdot n \cdot (B_{k-1,n-1}(x) - B_{k,n-1}(x))$$

$$= n \left(\sum_{k=0}^{n-1} \lambda_{k+1} B_{k,n-1}(x) - \sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) \right)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials Form a Basis

Lemma 97

The $n+1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Proof: We do a proof by induction.
I.B.: The claim is obviously true for $n := 0$ and $n := 1$.
I.H.: Suppose that the claim is true for an arbitrary but fixed $n-1 \in \mathbb{N}_0$, i.e., that $\sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) = 0$ implies $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = 0$.
I.S.: Suppose that $\sum_{k=0}^{n} \lambda_k B_{k,n}(x) = 0$ for some $\lambda_0, \lambda_1, \ldots, \lambda_n \in \mathbb{R}$. Then we get

$$
\begin{aligned}
0 = \sum_{k=0}^{n} \lambda_k B'_{k,n}(x) &\overset{\text{Lem. } 95}{=} \sum_{k=0}^{n} \lambda_k \cdot n \cdot (B_{k-1,n-1}(x) - B_{k,n-1}(x)) \\
&= n \left(\sum_{k=0}^{n-1} \lambda_{k+1} B_{k,n-1}(x) - \sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) \right) \\
&= n \sum_{k=0}^{n-1} \mu_k B_{k,n-1}(x) \qquad \text{with } \mu_k := \lambda_{k+1} - \lambda_k \text{ for } 0 \le k \le n-1.
\end{aligned}
$$

Bernstein Basis Polynomials Form a Basis

Lemma 97

The $n+1$ Bernstein basis polynomials $B_{0,n}, B_{1,n}, \ldots, B_{n,n}$ are linearly independent, for all $n \in \mathbb{N}_0$.

Proof : We do a proof by induction.
I.B.: The claim is obviously true for $n := 0$ and $n := 1$.
I.H.: Suppose that the claim is true for an arbitrary but fixed $n-1 \in \mathbb{N}_0$, i.e., that $\sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) = 0$ implies $\lambda_0 = \lambda_1 = \ldots = \lambda_{n-1} = 0$.
I.S.: Suppose that $\sum_{k=0}^{n} \lambda_k B_{k,n}(x) = 0$ for some $\lambda_0, \lambda_1, \ldots, \lambda_n \in \mathbb{R}$. Then we get

$$
\begin{aligned}
0 = \sum_{k=0}^{n} \lambda_k B'_{k,n}(x) &\overset{\text{Lem. 95}}{=} \sum_{k=0}^{n} \lambda_k \cdot n \cdot (B_{k-1,n-1}(x) - B_{k,n-1}(x)) \\
&= n \left(\sum_{k=0}^{n-1} \lambda_{k+1} B_{k,n-1}(x) - \sum_{k=0}^{n-1} \lambda_k B_{k,n-1}(x) \right) \\
&= n \sum_{k=0}^{n-1} \mu_k B_{k,n-1}(x) \qquad \text{with } \mu_k := \lambda_{k+1} - \lambda_k \text{ for } 0 \le k \le n-1.
\end{aligned}
$$

The I.H. implies $\mu_0 = \mu_1 = \cdots \mu_{n-1} = 0$ and, thus, $\lambda_0 = \lambda_1 = \cdots \lambda_n$, which implies $\lambda_0 = \lambda_1 = \cdots \lambda_n = 0$. (Recall Partition of Unity, Lem. 92.)

Bernstein Basis Polynomials Form a Basis

Lemma 98

For all $n, i \in \mathbb{N}_0$ with $i \leq n$, we have

$$x^i = \sum_{k=i}^{n} \frac{\binom{k}{i}}{\binom{n}{i}} B_{k,n}(x).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials Form a Basis

Lemma 98

For all $n, i \in \mathbb{N}_0$ with $i \leq n$, we have

$$x^i = \sum_{k=i}^{n} \frac{\binom{k}{i}}{\binom{n}{i}} B_{k,n}(x).$$

Theorem 99

The Bernstein basis polynomials of degree n form a basis of the vector space of polynomials of degree up to n over $\mathbb{R}$, for all $n \in \mathbb{N}_0$.

Bernstein Basis Polynomials Form a Basis

Lemma 98

For all $n, i \in \mathbb{N}_0$ with $i \leq n$, we have

$$x^i = \sum_{k=i}^{n} \frac{\binom{k}{i}}{\binom{n}{i}} B_{k,n}(x).$$

Theorem 99

The Bernstein basis polynomials of degree n form a basis of the vector space of polynomials of degree up to n over $\mathbb{R}$, for all $n \in \mathbb{N}_0$.

Proof : This is an immediate consequence of either Lem. 97 or Lem. 98. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

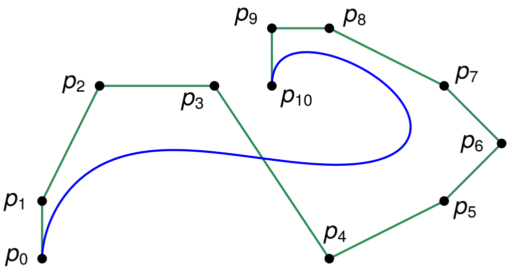
- Discovered in the late 1950s by Paul de Faget de Casteljau at Citroën and in the early 1960s by Pierre E. Bézier at Renault, and first published by Bézier in 1962. (Citroën allowed de Casteljau to publish his results in 1974 for the first time.)

Bézier Curves

- Discovered in the late 1950s by Paul de Faget de Casteljau at Citroën and in the early 1960s by Pierre E. Bézier at Renault, and first published by Bézier in 1962. (Citroën allowed de Casteljau to publish his results in 1974 for the first time.)
- The idea is to specify a curve by using points which control its shape: *control points*.

Bézier Curves

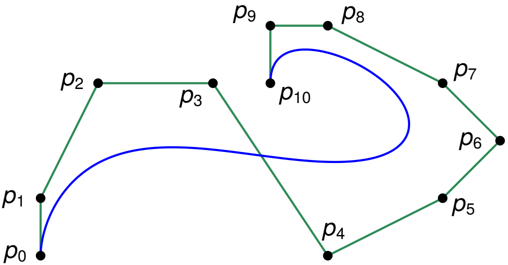
- Discovered in the late 1950s by Paul de Faget de Casteljau at Citroën and in the early 1960s by Pierre E. Bézier at Renault, and first published by Bézier in 1962. (Citroën allowed de Casteljau to publish his results in 1974 for the first time.)
- The idea is to specify a curve by using points which control its shape: *control points*. The figure shows a Bézier curve of degree 10 with 11 control points.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bézier Curves

- Discovered in the late 1950s by Paul de Faget de Casteljau at Citroën and in the early 1960s by Pierre E. Bézier at Renault, and first published by Bézier in 1962. (Citroën allowed de Casteljau to publish his results in 1974 for the first time.)
- The idea is to specify a curve by using points which control its shape: *control points*. The figure shows a Bézier curve of degree 10 with 11 control points.



- Bézier curves formed the foundations of the UNISURF CAD/CAM system.
- TrueType fonts use font descriptions made of composite quadratic Bézier curves; PostScript, METAFONT, and SVG use composite cubic Bézier curves.

Definition 100 (Bézier curve)

Suppose that we are given $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane $\mathbb{R}^2$, for $n \in \mathbb{N}$. The *Bézier curve* $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ defined by $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{B}(t) := \sum_{i=0}^{n} B_{i,n}(t) p_i \qquad \text{for } t \in [0,1],$$

where $B_{i,n}(t) := \binom{n}{i} t^i (1-t)^{n-i}$ is the i-th Bernstein basis polynomial of degree n.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 100 (Bézier curve)

Suppose that we are given $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane $\mathbb{R}^2$, for $n \in \mathbb{N}$. The *Bézier curve* $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ defined by $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{B}(t) := \sum_{i=0}^{n} B_{i,n}(t) p_i \qquad \text{for } t \in [0,1],$$

where $B_{i,n}(t) := \binom{n}{i} t^i (1-t)^{n-i}$ is the i-th Bernstein basis polynomial of degree n.

- The weighted average of all control points gives a location on the curve relative to the parameter t. The weights are given by the coefficients $B_{i,n}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 100 (Bézier curve)

Suppose that we are given $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane $\mathbb{R}^2$, for $n \in \mathbb{N}$. The *Bézier curve* $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ defined by $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{B}(t) := \sum_{i=0}^{n} B_{i,n}(t) p_i \qquad \text{for } t \in [0,1],$$

where $B_{i,n}(t) := \binom{n}{i} t^i (1-t)^{n-i}$ is the i-th Bernstein basis polynomial of degree n.

- The weighted average of all control points gives a location on the curve relative to the parameter t. The weights are given by the coefficients $B_{i,n}$.
- The polygonal chain $p_0, p_1, p_2, \ldots, p_{n-1}, p_n$ is called the *control polygon*, and its individual segments are referred to as *legs*.
- Although not explicitly required, it is generally assumed that the control points are distinct, except for possibly p_0 and p_n being identical.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bézier Curves

Definition 100 (Bézier curve)

Suppose that we are given $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane $\mathbb{R}^2$, for $n \in \mathbb{N}$. The *Bézier curve* $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ defined by $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{B}(t) := \sum_{i=0}^{n} B_{i,n}(t) p_i \qquad \text{for } t \in [0,1],$$

where $B_{i,n}(t) := \binom{n}{i} t^i (1-t)^{n-i}$ is the i-th Bernstein basis polynomial of degree n.

- The weighted average of all control points gives a location on the curve relative to the parameter t. The weights are given by the coefficients $B_{i,n}$.
- The polygonal chain $p_0, p_1, p_2, \ldots, p_{n-1}, p_n$ is called the *control polygon*, and its individual segments are referred to as *legs*.
- Although not explicitly required, it is generally assumed that the control points are distinct, except for possibly p_0 and p_n being identical.
- Of course, the same definition and the subsequent math can be applied to $p_0, p_1, \ldots, p_n \in \mathbb{R}^d$ for some $d \in \mathbb{N}$ with $d > 2$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Curves

Lemma 101

A Bézier curve defined by $n + 1$ control points is (coordinate-wise) a polynomial of degree n.

Lemma 101

A Bézier curve defined by $n + 1$ control points is (coordinate-wise) a polynomial of degree n.

Proof : It is the sum of $n + 1$ Bernstein basis polynomials of degree n. □

Lemma 101

A Bézier curve defined by $n+1$ control points is (coordinate-wise) a polynomial of degree n.

Proof : It is the sum of $n+1$ Bernstein basis polynomials of degree n. □

Lemma 102

A Bézier curve starts in the first control point and ends in the last control point.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 101

A Bézier curve defined by $n+1$ control points is (coordinate-wise) a polynomial of degree n.

Proof: It is the sum of $n+1$ Bernstein basis polynomials of degree n. □

Lemma 102

A Bézier curve starts in the first control point and ends in the last control point.

Proof: Recall that

$$B_{i,n}(0) = \binom{n}{i} 0^i (1-0)^{n-i} = \begin{cases} 1 & \text{for } i = 0, \\ 0 & \text{for } i > 0. \end{cases}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Curves

Lemma 101

A Bézier curve defined by $n+1$ control points is (coordinate-wise) a polynomial of degree n.

Proof: It is the sum of $n+1$ Bernstein basis polynomials of degree n. □

Lemma 102

A Bézier curve starts in the first control point and ends in the last control point.

Proof: Recall that

$$B_{i,n}(0) = \binom{n}{i} 0^i \, (1-0)^{n-i} = \begin{cases} 1 & \text{for } i = 0, \\ 0 & \text{for } i > 0. \end{cases}$$

Hence,

$$\mathcal{B}(0) = \sum_{i=0}^{n} B_{i,n}(0) p_i = B_{0,n}(0) p_0 = p_0.$$

Similarly for $B_{i,n}(1)$ and $\mathcal{B}(1)$.

Lemma 103 (Convex hull property)

A Bézier curve lies completely inside the convex hull of its control points.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 103 (Convex hull property)

A Bézier curve lies completely inside the convex hull of its control points.

Proof : This is nothing but a re-formulation of Cor. 94. □

Properties of Bézier Curves

Lemma 103 (Convex hull property)

A Bézier curve lies completely inside the convex hull of its control points.

Proof : This is nothing but a re-formulation of Cor. 94. □

Lemma 104 (Variation diminishing property)

If a straight line intersects the control polygon of a Bézier curve k times then it intersects the actual Bézier curve at most k times.

Properties of Bézier Curves

Lemma 103 (Convex hull property)

A Bézier curve lies completely inside the convex hull of its control points.

Proof: This is nothing but a re-formulation of Cor. 94. □

Lemma 104 (Variation diminishing property)

If a straight line intersects the control polygon of a Bézier curve k times then it intersects the actual Bézier curve at most k times.

Lemma 105 (Symmetry property)

The following identity holds for all $n \in \mathbb{N}$, all $p_0, \ldots, p_n \in \mathbb{R}^2$ and all $t \in [0, 1]$:

$$\sum_{i=0}^{n} B_{i,n}(t) p_i = \sum_{i=0}^{n} B_{i,n}(1-t) p_{n-i}.$$

Lemma 106 (Affine invariance)

Any Bézier representation is affinely invariant, i.e., given any affine map π, the image curve $\pi(\mathcal{B})$ of a Bézier curve $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$ over $[0,1]$.

Lemma 106 (Affine invariance)

Any Bézier representation is affinely invariant, i.e., given any affine map π, the image curve $\pi(\mathcal{B})$ of a Bézier curve $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$ over $[0,1]$.

Proof : Consider an affine map $\pi\colon \mathbb{R}^2 \to \mathbb{R}^2$. Hence, $\pi(x) = \mathbf{A} \cdot x + v$, for some 2×2 matrix $\mathbf{A}$, and $x, v \in \mathbb{R}^2$.

Lemma 106 (Affine invariance)

Any Bézier representation is affinely invariant, i.e., given any affine map π, the image curve $\pi(\mathcal{B})$ of a Bézier curve $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$ over $[0,1]$.

Proof: Consider an affine map $\pi\colon \mathbb{R}^2 \to \mathbb{R}^2$. Hence, $\pi(x) = \mathbf{A} \cdot x + v$, for some 2×2 matrix $\mathbf{A}$, and $x, v \in \mathbb{R}^2$. We get

$$\pi(\mathcal{B}(t)) = \pi\left(\sum_{i=0}^{n} B_{i,n}(t) p_i\right) = \mathbf{A} \cdot \left(\sum_{i=0}^{n} B_{i,n}(t) p_i\right) + v$$

Lemma 106 (Affine invariance)

Any Bézier representation is affinely invariant, i.e., given any affine map π, the image curve $\pi(\mathcal{B})$ of a Bézier curve $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$ over $[0,1]$.

Proof: Consider an affine map $\pi\colon \mathbb{R}^2 \to \mathbb{R}^2$. Hence, $\pi(x) = \mathbf{A} \cdot x + v$, for some 2×2 matrix $\mathbf{A}$, and $x, v \in \mathbb{R}^2$. We get

$$\pi(\mathcal{B}(t)) = \pi\left(\sum_{i=0}^{n} B_{i,n}(t) p_i\right) = \mathbf{A} \cdot \left(\sum_{i=0}^{n} B_{i,n}(t) p_i\right) + v$$

$$= \sum_{i=0}^{n} B_{i,n}(t) \mathbf{A} \cdot p_i + \sum_{i=0}^{n} B_{i,n}(t) v = \sum_{i=0}^{n} B_{i,n}(t)(\mathbf{A} \cdot p_i + v)$$

Lemma 106 (Affine invariance)

Any Bézier representation is affinely invariant, i.e., given any affine map π, the image curve $\pi(\mathcal{B})$ of a Bézier curve $\mathcal{B}\colon [0,1] \to \mathbb{R}^2$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$ over $[0,1]$.

Proof: Consider an affine map $\pi\colon \mathbb{R}^2 \to \mathbb{R}^2$. Hence, $\pi(x) = \mathbf{A} \cdot x + v$, for some 2×2 matrix $\mathbf{A}$, and $x, v \in \mathbb{R}^2$. We get

$$\pi(\mathcal{B}(t)) = \pi\left(\sum_{i=0}^{n} B_{i,n}(t) p_i\right) = \mathbf{A} \cdot \left(\sum_{i=0}^{n} B_{i,n}(t) p_i\right) + v$$

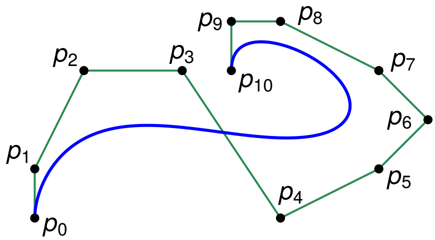
$$= \sum_{i=0}^{n} B_{i,n}(t) \mathbf{A} \cdot p_i + \sum_{i=0}^{n} B_{i,n}(t) v = \sum_{i=0}^{n} B_{i,n}(t)(\mathbf{A} \cdot p_i + v)$$

$$= \sum_{i=0}^{n} B_{i,n}(t) \pi(p_i).$$

□

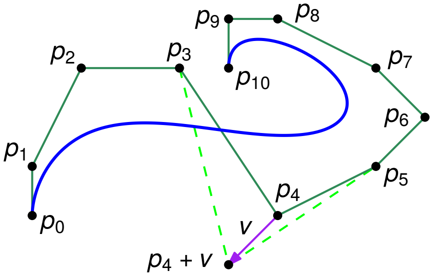
Modifying a Control Point

- Suppose that we shift one control point p_j to a new location $p_j + v$.



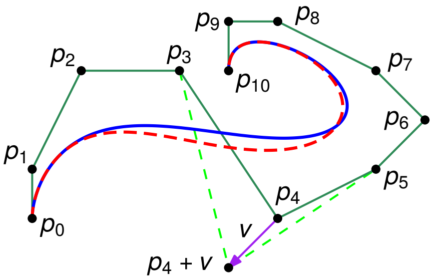
Modifying a Control Point

- Suppose that we shift one control point p_j to a new location $p_j + v$.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Suppose that we shift one control point p_j to a new location $p_j + v$.



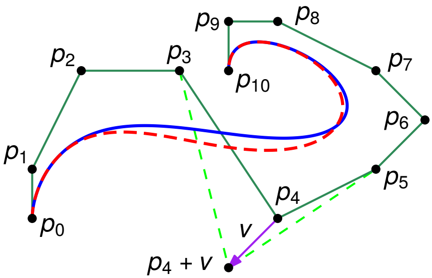
- The corresponding Bézier curve $\mathcal{B}$ is transformed to $\mathcal{B}^\star$ as follows:

$$\mathcal{B}^\star(t) = \left(\sum_{i=0}^{j-1} B_{i,n}(t)p_i\right) + B_{j,n}(t)(p_j + v) + \left(\sum_{i=j+1}^{n} B_{i,n}(t)p_i\right) =$$

$$= \sum_{i=0}^{n} B_{i,n}(t)p_i + B_{j,n}(t)v = \mathcal{B}(t) + B_{j,n}(t)v$$

Modifying a Control Point

- Suppose that we shift one control point p_j to a new location $p_j + v$.



- The corresponding Bézier curve $\mathcal{B}$ is transformed to $\mathcal{B}^\star$ as follows:

$$\mathcal{B}^\star(t) = \left(\sum_{i=0}^{j-1} B_{i,n}(t)p_i\right) + B_{j,n}(t)(p_j + v) + \left(\sum_{i=j+1}^{n} B_{i,n}(t)p_i\right) =$$

$$= \sum_{i=0}^{n} B_{i,n}(t)p_i + B_{j,n}(t)v = \mathcal{B}(t) + B_{j,n}(t)v$$

- Now recall that $B_{j,n}(t) \neq 0$ for all t with $0 < t < 1$. Hence, a modification of just one control point results in a global change of the entire Bézier curve.

Evaluation of a Bézier Curve

- For $0 < t < 1$ we can locate a point q on a line segment $\overline{pr}$ such that it divides the line segment into portions of relative length t and $1 - t$, i.e., according to the ratio $t : (1 - t)$.
- Of course, q is given by the linear interpolation

$$q = p + t(r - p) = (1 - t) \cdot p + t \cdot r.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a Bézier Curve

- For $0 < t < 1$ we can locate a point q on a line segment $\overline{pr}$ such that it divides the line segment into portions of relative length t and $1 - t$, i.e., according to the ratio $t : (1 - t)$.
- Of course, q is given by the linear interpolation

$$q = p + t(r - p) = (1 - t) \cdot p + t \cdot r.$$

- Similarly, we can compute a point on a Bézier curve such that the curve is split into portions of relative length t and $1 - t$.
  - On every leg $\overline{p_{j-1}p_j}$ of the control polygon we compute a point p_{1j} which divides it according to the ratio $t : (1 - t)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a Bézier Curve

- For $0 < t < 1$ we can locate a point q on a line segment $\overline{pr}$ such that it divides the line segment into portions of relative length t and $1 - t$, i.e., according to the ratio $t : (1 - t)$.
- Of course, q is given by the linear interpolation

$$q = p + t(r - p) = (1 - t) \cdot p + t \cdot r.$$

- Similarly, we can compute a point on a Bézier curve such that the curve is split into portions of relative length t and $1 - t$.
  - On every leg $\overline{p_{j-1}p_j}$ of the control polygon we compute a point p_{1j} which divides it according to the ratio $t : (1 - t)$.
  - In total we get n new points which define a new polygonal chain with $n - 1$ legs.
  - This new polygonal chain can be used to construct another polygonal chain with $n - 2$ legs.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a Bézier Curve

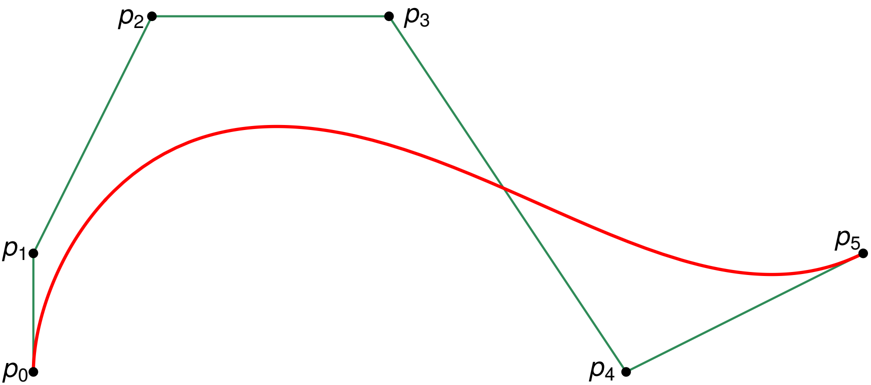
- For $0 < t < 1$ we can locate a point q on a line segment $\overline{pr}$ such that it divides the line segment into portions of relative length t and $1 - t$, i.e., according to the ratio $t : (1 - t)$.
- Of course, q is given by the linear interpolation

$$q = p + t(r - p) = (1 - t) \cdot p + t \cdot r.$$

- Similarly, we can compute a point on a Bézier curve such that the curve is split into portions of relative length t and $1 - t$.
  - On every leg $\overline{p_{j-1}p_j}$ of the control polygon we compute a point p_{1j} which divides it according to the ratio $t : (1 - t)$.
  - In total we get n new points which define a new polygonal chain with $n - 1$ legs.
  - This new polygonal chain can be used to construct another polygonal chain with $n - 2$ legs.
  - This process can be repeated n times, i.e., until we are left with a single point.
  - It was proved by de Casteljau that this point corresponds to the point $\mathcal{B}(t)$ sought.

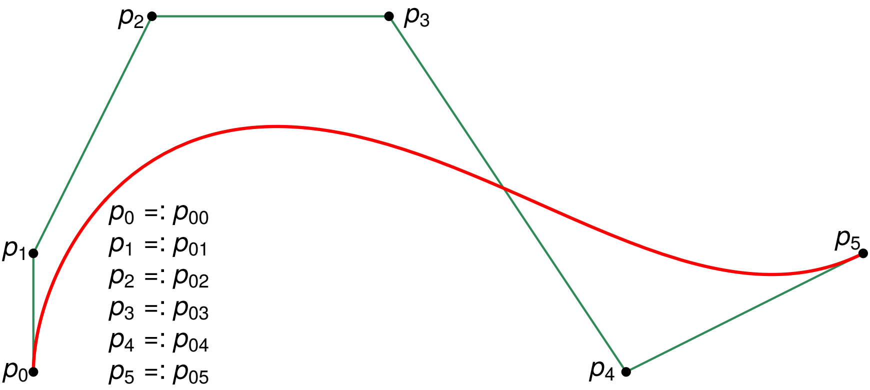
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Sample run of de Casteljau's algorithm for $t := 1/4$.



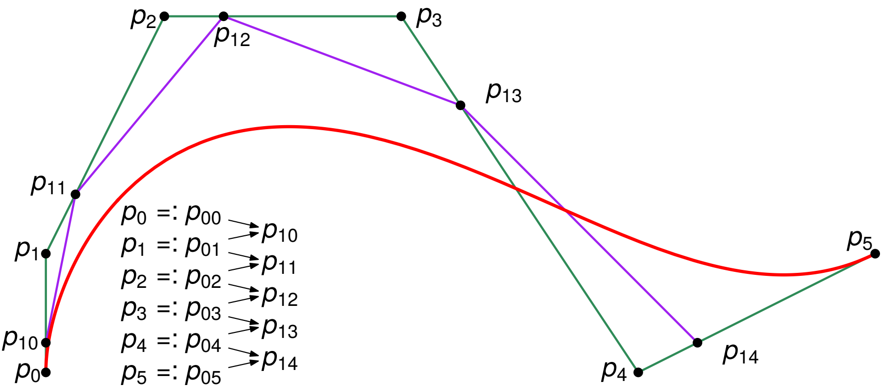
De Casteljau's Algorithm

- Sample run of de Casteljau's algorithm for $t := 1/4$.
- The points are indexed in the form i, j, where i denotes the number of the iteration and $j + 1$ numbers the leg defined by the control points $p_{i,j}$ and $p_{i,j+1}$.

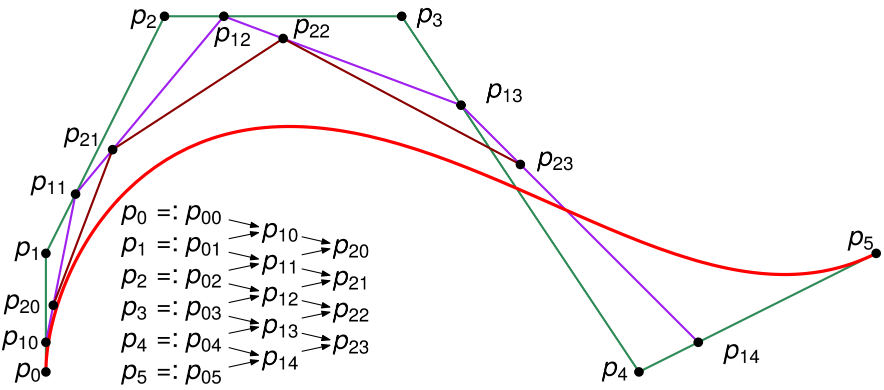


De Casteljau's Algorithm

- Sample run of de Casteljau's algorithm for $t := 1/4$.
- The points are indexed in the form i, j, where i denotes the number of the iteration and $j+1$ numbers the leg defined by the control points $p_{i,j}$ and $p_{i,j+1}$.

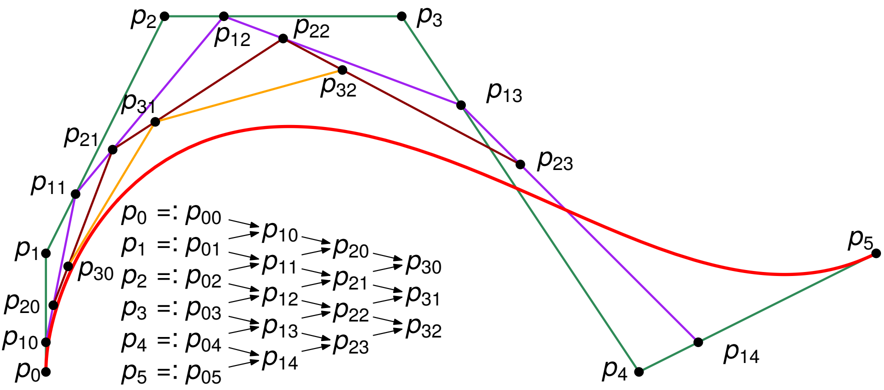


- Sample run of de Casteljau's algorithm for $t := 1/4$.
- The points are indexed in the form i, j, where i denotes the number of the iteration and $j + 1$ numbers the leg defined by the control points $p_{i,j}$ and $p_{i,j+1}$.



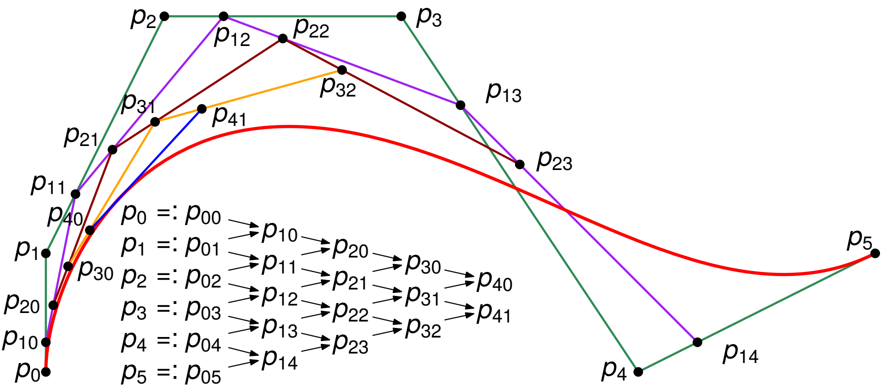
De Casteljau's Algorithm

- Sample run of de Casteljau's algorithm for $t := 1/4$.
- The points are indexed in the form i, j, where i denotes the number of the iteration and $j+1$ numbers the leg defined by the control points $p_{i,j}$ and $p_{i,j+1}$.



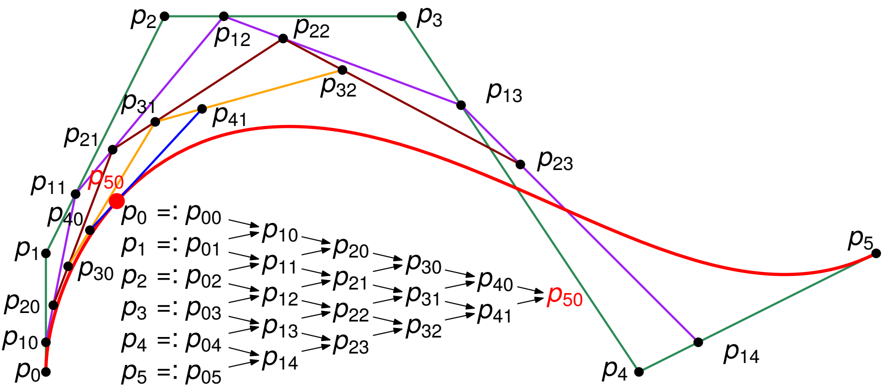
De Casteljau's Algorithm

- Sample run of de Casteljau's algorithm for $t := 1/4$.
- The points are indexed in the form i, j, where i denotes the number of the iteration and $j + 1$ numbers the leg defined by the control points $p_{i,j}$ and $p_{i,j+1}$.



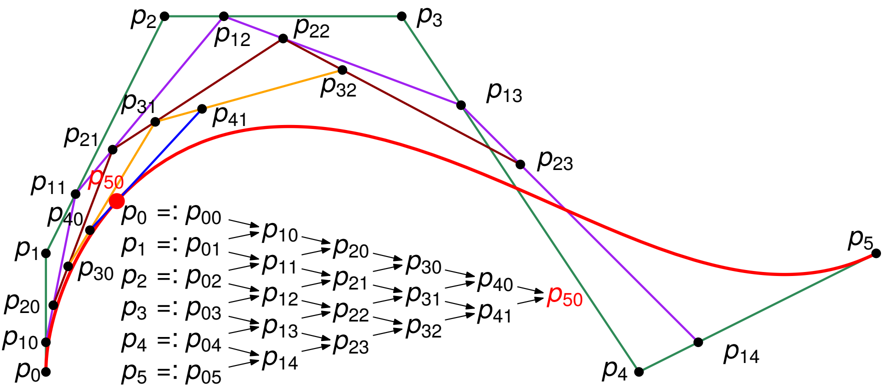
De Casteljau's Algorithm

- Sample run of de Casteljau's algorithm for $t := 1/4$.
- The points are indexed in the form i, j, where i denotes the number of the iteration and $j+1$ numbers the leg defined by the control points $p_{i,j}$ and $p_{i,j+1}$.



De Casteljau's Algorithm

- Sample run of de Casteljau's algorithm for $t := 1/4$.
- The points are indexed in the form i, j, where i denotes the number of the iteration and $j + 1$ numbers the leg defined by the control points $p_{i,j}$ and $p_{i,j+1}$.



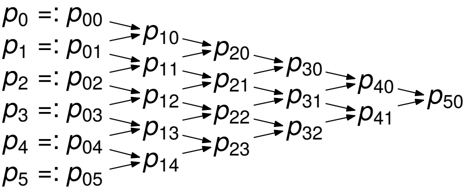
- The union of these legs $\overline{p_{i,j}, p_{i,j+1}}$ is known as *de Casteljau net*.

De Casteljau's Algorithm

- Numerically very stable, since only convex combinations are used!

```
/** Evaluates a Bezier curve at parameter t by applying de Casteljau's algorithm
 * @param p: array of n+1 control points
 * @param n: the degree of the Bezier curve
 * @param t: the parameter
 * @return the evaluation result
 */
point DeCasteljau(point *p, int n, double t)
{
  for (int i = 1;  i <= n;  ++i)
      for (int j = 0;  j <= n-i;  ++j)
           p[j] = (1-t) * p[j] + t * p[j+1];

  return p[0];
}
```



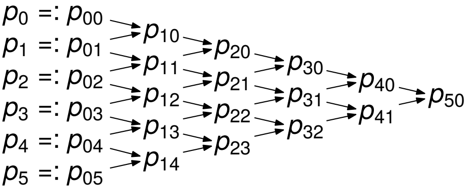
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

De Casteljau's Algorithm: Correctness

- The point p_{10} is obtained as

$$p_{10} = (1-t) \cdot p_{00} + t \cdot p_{01}.$$

- Hence, the contribution of p_{01} to p_{10} is $t \cdot p_{01}$.



- The point p_{10} is obtained as

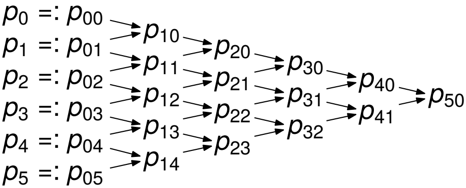
$$p_{10} = (1-t) \cdot p_{00} + t \cdot p_{01}.$$

- Hence, the contribution of p_{01} to p_{10} is $t \cdot p_{01}$.
- Since p_{20} is obtained as

$$p_{20} = (1-t) \cdot p_{10} + t \cdot p_{11},$$

the contribution of p_{01} to p_{20} via p_{10} is

$$(1-t)p_{10} = t(1-t) \cdot p_{01}.$$



De Casteljau's Algorithm: Correctness

- The point p_{10} is obtained as

$$p_{10} = (1-t) \cdot p_{00} + t \cdot p_{01}.$$

- Hence, the contribution of p_{01} to p_{10} is $t \cdot p_{01}$.
- Since p_{20} is obtained as

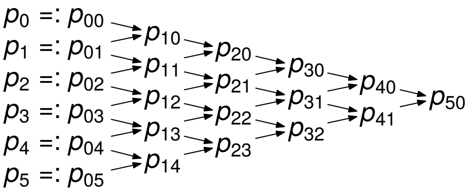
$$p_{20} = (1-t) \cdot p_{10} + t \cdot p_{11},$$

  the contribution of p_{01} to p_{20} via p_{10} is

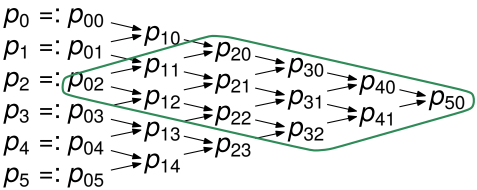
$$(1-t)p_{10} = t(1-t) \cdot p_{01}.$$

- Similarly, the contribution of p_{01} to p_{20} via p_{11} is

$$t(1-t) \cdot p_{01}.$$

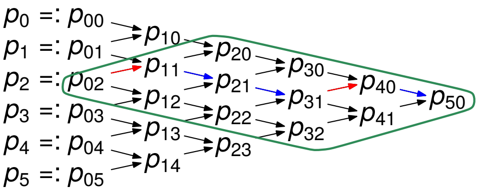


- Each path from p_{0i} to p_{n0} is constrained to a diamond shape anchored at p_{0i} and p_{n0}.



De Casteljau's Algorithm: Correctness

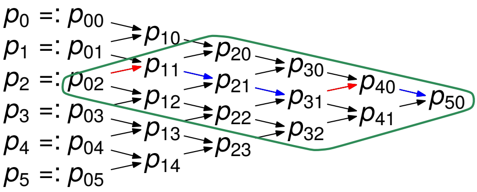
- Each path from p_{0i} to p_{n0} is constrained to a diamond shape anchored at p_{0i} and p_{n0}.
- An inductive argument shows that *each* path from p_{0i} to p_{n0} consists of i north-east arrows, i.e., multiplications by t, and $n - i$ south-east arrows, i.e., multiplications by $(1 - t)$.



- Each path from p_{0i} to p_{n0} is constrained to a diamond shape anchored at p_{0i} and p_{n0}.
- An inductive argument shows that *each* path from p_{0i} to p_{n0} consists of i north-east arrows, i.e., multiplications by t, and $n - i$ south-east arrows, i.e., multiplications by $(1 - t)$.
- Thus, the contribution of p_{0i} to p_{n0} is

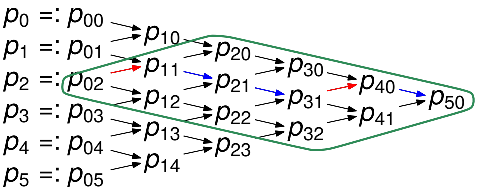
$$t^i(1 - t)^{n-i} \cdot p_{0i},$$

along *each* path from p_{0i} to p_{n0}.



De Casteljau's Algorithm: Correctness

- How many different paths exist from p_{0i} to p_{n0}? This is equivalent to asking "how many different ways exist to place i north-east arrows on a total of n possible positions?", and the answer is given by $\binom{n}{i}$.



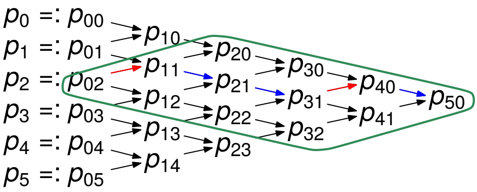
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

De Casteljau's Algorithm: Correctness

- How many different paths exist from p_{0i} to p_{n0}? This is equivalent to asking "how many different ways exist to place i north-east arrows on a total of n possible positions?", and the answer is given by $\binom{n}{i}$.
- Thus, the total contribution of p_{0i} to p_{n0}, along all paths from p_{0i} to p_{n0}, is

$$\binom{n}{i} \cdot t^i (1-t)^{n-i} p_{0i}.$$

This is, however, precisely the weight of p_{0i}, i.e., p_i in the definition of a Bézier curve (Def. 100).



Evaluation of a Bézier Curve Using Horner's Scheme

- Horner's scheme can also be used for evaluating a Bézier curve.
- After rewriting $\mathcal{B}(t)$ as

$$\mathcal{B}(t) = \sum_{i=0}^{n} B_{i,n}(t) p_i = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} p_i$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a Bézier Curve Using Horner's Scheme

- Horner's scheme can also be used for evaluating a Bézier curve.
- After rewriting $\mathcal{B}(t)$ as

$$\begin{aligned}\mathcal{B}(t) = \sum_{i=0}^{n} B_{i,n}(t)p_i = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} p_i \\ = (1-t)^n \left(\sum_{i=0}^{n} \binom{n}{i} \left(\frac{t}{1-t} \right)^i p_i \right),\end{aligned}$$

one evaluates the sum for the value $\frac{t}{1-t}$, and then multiplies by $(1-t)^n$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a Bézier Curve Using Horner's Scheme

- Horner's scheme can also be used for evaluating a Bézier curve.
- After rewriting $\mathcal{B}(t)$ as

$$\begin{aligned}\mathcal{B}(t) = \sum_{i=0}^{n} B_{i,n}(t)p_i &= \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} p_i \\ &= (1-t)^n \left(\sum_{i=0}^{n} \binom{n}{i} \left(\frac{t}{1-t}\right)^i p_i \right),\end{aligned}$$

  one evaluates the sum for the value $\frac{t}{1-t}$, and then multiplies by $(1-t)^n$.
- This method becomes unstable if t is close to one. In this case, one can resort to Lem. 105, which gives the identity

$$\mathcal{B}(t) = t^n \left(\sum_{i=0}^{n} \binom{n}{i} \left(\frac{1-t}{t}\right)^i p_{n-i} \right).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a Bézier Curve Using Horner's Scheme

- Horner's scheme can also be used for evaluating a Bézier curve.
- After rewriting $\mathcal{B}(t)$ as

$$\mathcal{B}(t) = \sum_{i=0}^{n} B_{i,n}(t) p_i = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} p_i$$
$$= (1-t)^n \left(\sum_{i=0}^{n} \binom{n}{i} \left(\frac{t}{1-t} \right)^i p_i \right),$$

  one evaluates the sum for the value $\frac{t}{1-t}$, and then multiplies by $(1-t)^n$.
- This method becomes unstable if t is close to one. In this case, one can resort to Lem. 105, which gives the identity

$$\mathcal{B}(t) = t^n \left(\sum_{i=0}^{n} \binom{n}{i} \left(\frac{1-t}{t} \right)^i p_{n-i} \right).$$

- In any case, Horner's scheme tends to be faster but numerically more problematic than de Casteljau's algorithm.

Evaluation of a Bézier Curve Using Horner's Scheme

- Horner's scheme can also be used for evaluating a Bézier curve.
- After rewriting $\mathcal{B}(t)$ as

$$\mathcal{B}(t) = \sum_{i=0}^{n} B_{i,n}(t) p_i = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} p_i$$
$$= (1-t)^n \left(\sum_{i=0}^{n} \binom{n}{i} \left(\frac{t}{1-t} \right)^i p_i \right),$$

  one evaluates the sum for the value $\frac{t}{1-t}$, and then multiplies by $(1-t)^n$.
- This method becomes unstable if t is close to one. In this case, one can resort to Lem. 105, which gives the identity

$$\mathcal{B}(t) = t^n \left(\sum_{i=0}^{n} \binom{n}{i} \left(\frac{1-t}{t} \right)^i p_{n-i} \right).$$

- In any case, Horner's scheme tends to be faster but numerically more problematic than de Casteljau's algorithm.
- [Woźny&Chudy (2019)] explain an algorithm that uses only convex combinations of the control points and consumes $O(n)$ time.

Theorem 107

Let $n, d \in \mathbb{N}$. For every polynomial function $F\colon \mathbb{R} \to \mathbb{R}^d$ of degree at most n there exists exactly one symmetric and multi-affine function $f\colon \mathbb{R}^n \to \mathbb{R}^d$ such that

Theorem 107

Let $n, d \in \mathbb{N}$. For every polynomial function $F\colon \mathbb{R} \to \mathbb{R}^d$ of degree at most n there exists exactly one symmetric and multi-affine function $f\colon \mathbb{R}^n \to \mathbb{R}^d$ such that

1. for all $i \in \{1, 2, \dots, n\}$, all $x_1, x_2, \dots, x_n \in \mathbb{R}$, all $k \in \mathbb{N}$, all $y_1, y_2, \dots, y_k \in \mathbb{R}$ and all $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{R}$ with $\sum_{j=1}^{k} \alpha_j = 1$

$$f(x_1, \dots, x_{i-1}, \sum_{j=1}^{k} \alpha_j y_j, x_{i+1}, \dots, x_n) = \sum_{j=1}^{k} \alpha_j f(x_1, \dots, x_{i-1}, y_j, x_{i+1}, \dots, x_n)$$

Theorem 107

Let $n, d \in \mathbb{N}$. For every polynomial function $F\colon \mathbb{R} \to \mathbb{R}^d$ of degree at most n there exists exactly one symmetric and multi-affine function $f\colon \mathbb{R}^n \to \mathbb{R}^d$ such that

1. for all $i \in \{1, 2, \ldots, n\}$, all $x_1, x_2, \ldots, x_n \in \mathbb{R}$, all $k \in \mathbb{N}$, all $y_1, y_2, \ldots, y_k \in \mathbb{R}$ and all $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{R}$ with $\sum_{j=1}^{k} \alpha_j = 1$

$$f(x_1, \ldots, x_{i-1}, \sum_{j=1}^{k} \alpha_j y_j, x_{i+1}, \ldots, x_n) = \sum_{j=1}^{k} \alpha_j f(x_1, \ldots, x_{i-1}, y_j, x_{i+1}, \ldots, x_n)$$

2. for all $i, j \in \{1, 2, \ldots, n\}$

$$\begin{aligned} f(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_n) &= \\ f(x_1, x_2, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_n), & \end{aligned}$$

Theorem 107

Let $n, d \in \mathbb{N}$. For every polynomial function $F\colon \mathbb{R} \to \mathbb{R}^d$ of degree at most n there exists exactly one symmetric and multi-affine function $f\colon \mathbb{R}^n \to \mathbb{R}^d$ such that

1. for all $i \in \{1, 2, \ldots, n\}$, all $x_1, x_2, \ldots, x_n \in \mathbb{R}$, all $k \in \mathbb{N}$, all $y_1, y_2, \ldots, y_k \in \mathbb{R}$ and all $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{R}$ with $\sum_{j=1}^{k} \alpha_j = 1$
$$f(x_1, \ldots, x_{i-1}, \sum_{j=1}^{k} \alpha_j y_j, x_{i+1}, \ldots, x_n) = \sum_{j=1}^{k} \alpha_j f(x_1, \ldots, x_{i-1}, y_j, x_{i+1}, \ldots, x_n)$$
2. for all $i, j \in \{1, 2, \ldots, n\}$
$$\begin{aligned} f(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_n) \; = \\ f(x_1, x_2, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_n), \end{aligned}$$
3. for all $x \in \mathbb{R}$
$$F(x) = f(\underbrace{x, x, \ldots, x}_{n \text{ times}}), \qquad \text{i.e., } F \text{ is the diagonal of } f.$$

Theorem 107

Let $n, d \in \mathbb{N}$. For every polynomial function $F\colon \mathbb{R} \to \mathbb{R}^d$ of degree at most n there exists exactly one symmetric and multi-affine function $f\colon \mathbb{R}^n \to \mathbb{R}^d$ such that

1. for all $i \in \{1, 2, \ldots, n\}$, all $x_1, x_2, \ldots, x_n \in \mathbb{R}$, all $k \in \mathbb{N}$, all $y_1, y_2, \ldots, y_k \in \mathbb{R}$ and all $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{R}$ with $\sum_{j=1}^{k} \alpha_j = 1$

$$f(x_1, \ldots, x_{i-1}, \sum_{j=1}^{k} \alpha_j y_j, x_{i+1}, \ldots, x_n) = \sum_{j=1}^{k} \alpha_j f(x_1, \ldots, x_{i-1}, y_j, x_{i+1}, \ldots, x_n)$$

2. for all $i, j \in \{1, 2, \ldots, n\}$

$$\begin{aligned} f(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_n) &= \\ f(x_1, x_2, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_n), & \end{aligned}$$

3. for all $x \in \mathbb{R}$

$$F(x) = f(\underbrace{x, x, \ldots, x}_{n \text{ times}}), \qquad \text{i.e., } F \text{ is the diagonal of } f.$$

The function f is called the *polar form* (aka “blossom”, Dt.: Polarform) of F.

Lemma 108

Let $n \in \mathbb{N}$ and $a_0, a_1, \ldots, a_n \in \mathbb{R}$, and $F(x) := \sum_{i=0}^{n} a_i x^i$. Then $f\colon \mathbb{R}^n \to \mathbb{R}$ with

$$f(x_1, x_2, \ldots, x_n) := \sum_{i=0}^{n} a_i \frac{1}{\binom{n}{i}} \left(\sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I|=i}} \prod_{j \in I} x_j \right)$$

is the polar form of F.

Lemma 108

Let $n \in \mathbb{N}$ and $a_0, a_1, \ldots, a_n \in \mathbb{R}$, and $F(x) := \sum_{i=0}^{n} a_i x^i$. Then $f\colon \mathbb{R}^n \to \mathbb{R}$ with

$$f(x_1, x_2, \ldots, x_n) := \sum_{i=0}^{n} a_i \frac{1}{\binom{n}{i}} \left(\sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I|=i}} \prod_{j \in I} x_j \right)$$

is the polar form of F.

	a_i	$F(x)$	$f(x_1, \ldots, x_n)$
$n = 1$	$a_0 := 1, a_1 := 0$	1	1
	$a_0 := 0, a_1 := 1$	x	x_1
$n = 2$	$a_0 := 1, a_1 := 0, a_2 := 0$	1	1
	$a_0 := 0, a_1 := 1, a_2 := 0$	x	$\frac{1}{2}(x_1 + x_2)$
	$a_0 := 0, a_1 := 0, a_2 := 1$	x^2	$x_1 x_2$
$n = 3$	$a_0 := 1, a_1 := 0, a_2 := 0, a_3 := 0$	1	1
	$a_0 := 0, a_1 := 1, a_2 := 0, a_3 := 0$	x	$\frac{1}{3}(x_1 + x_2 + x_3)$
	$a_0 := 0, a_1 := 0, a_2 := 1, a_3 := 0$	x^2	$\frac{1}{3}(x_1 x_2 + x_1 x_3 + x_2 x_3)$
	$a_0 := 0, a_1 := 0, a_2 := 0, a_3 := 1$	x^3	$x_1 x_2 x_3$

Bernstein Polynomials and Polar Forms

- Let $F(x) := \begin{pmatrix} x \\ \frac{1}{2}x^2 \end{pmatrix}$. Hence $f(x_1, x_2) = \begin{pmatrix} \frac{1}{2}(x_1 + x_2) \\ \frac{1}{2}x_1 x_2 \end{pmatrix}$,

Bernstein Polynomials and Polar Forms

- Let $F(x) := \begin{pmatrix} x \\ \frac{1}{2}x^2 \end{pmatrix}$. Hence $f(x_1, x_2) = \begin{pmatrix} \frac{1}{2}(x_1 + x_2) \\ \frac{1}{2}x_1 x_2 \end{pmatrix}$, and we get

$$f(0,0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad f(0,1) = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} = f(1,0) \qquad f(1,1) = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

- Let $F(x) := \begin{pmatrix} x \\ \frac{1}{2}x^2 \end{pmatrix}$. Hence $f(x_1, x_2) = \begin{pmatrix} \frac{1}{2}(x_1 + x_2) \\ \frac{1}{2}x_1 x_2 \end{pmatrix}$, and we get

$$f(0,0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad f(0,1) = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} = f(1,0) \qquad f(1,1) = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

- Furthermore, $F(t) = f(t,t)$, with

$$\begin{aligned} F(t) &= f(t,t) \\ &= f((1-t)\cdot 0 + t\cdot 1, t) = (1-t)\cdot f(0,t) + t\cdot f(1,t) \end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Polynomials and Polar Forms

- Let $F(x) := \begin{pmatrix} x \\ \frac{1}{2}x^2 \end{pmatrix}$. Hence $f(x_1, x_2) = \begin{pmatrix} \frac{1}{2}(x_1 + x_2) \\ \frac{1}{2}x_1 x_2 \end{pmatrix}$, and we get

$$f(0,0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad f(0,1) = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} = f(1,0) \qquad f(1,1) = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

- Furthermore, $F(t) = f(t,t)$, with

$$\begin{aligned} F(t) &= f(t,t) \\ &= f((1-t)\cdot 0 + t \cdot 1, t) = (1-t)\cdot f(0,t) + t \cdot f(1,t) \\ &= (1-t)[(1-t)\cdot f(0,0) + t \cdot f(0,1))] + t[(1-t)\cdot f(1,0) + t \cdot f(1,1))] \\ &= (1-t)^2 \cdot f(0,0) + 2t(1-t)\cdot f(0,1) + t^2 \cdot f(1,1) \end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Let $F(x) := \begin{pmatrix} x \\ \frac{1}{2}x^2 \end{pmatrix}$. Hence $f(x_1, x_2) = \begin{pmatrix} \frac{1}{2}(x_1 + x_2) \\ \frac{1}{2}x_1 x_2 \end{pmatrix}$, and we get

$$f(0,0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad f(0,1) = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} = f(1,0) \qquad f(1,1) = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

- Furthermore, $F(t) = f(t,t)$, with

$$\begin{aligned} F(t) &= f(t,t) \\ &= f((1-t)\cdot 0 + t \cdot 1, t) = (1-t)\cdot f(0,t) + t \cdot f(1,t) \\ &= (1-t)[(1-t)\cdot f(0,0) + t\cdot f(0,1))] + t[(1-t)\cdot f(1,0) + t \cdot f(1,1))] \\ &= (1-t)^2 \cdot f(0,0) + 2t(1-t)\cdot f(0,1) + t^2 \cdot f(1,1) \\ &= B_{0,2}(t) f(0,0) + B_{1,2}(t) f(0,1) + B_{2,2}(t) f(1,1) \\ &= B_{0,2}(t)\begin{pmatrix} 0 \\ 0 \end{pmatrix} + B_{1,2}(t)\begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} + B_{2,2}(t)\begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}. \end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Let $F(x) := \begin{pmatrix} x \\ \frac{1}{2}x^2 \end{pmatrix}$. Hence $f(x_1, x_2) = \begin{pmatrix} \frac{1}{2}(x_1 + x_2) \\ \frac{1}{2}x_1 x_2 \end{pmatrix}$, and we get

$$f(0,0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad f(0,1) = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} = f(1,0) \qquad f(1,1) = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

- Furthermore, $F(t) = f(t,t)$, with

$$\begin{aligned} F(t) &= f(t,t) \\ &= f((1-t)\cdot 0 + t\cdot 1, t) = (1-t)\cdot f(0,t) + t\cdot f(1,t) \\ &= (1-t)[(1-t)\cdot f(0,0) + t\cdot f(0,1))] + t[(1-t)\cdot f(1,0) + t\cdot f(1,1))] \\ &= (1-t)^2\cdot f(0,0) + 2t(1-t)\cdot f(0,1) + t^2\cdot f(1,1) \\ &= B_{0,2}(t)f(0,0) + B_{1,2}(t)f(0,1) + B_{2,2}(t)f(1,1) \\ &= B_{0,2}(t)\begin{pmatrix} 0 \\ 0 \end{pmatrix} + B_{1,2}(t)\begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} + B_{2,2}(t)\begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}. \end{aligned}$$

- Hence, there is a close connection between the polar form and the Bernstein polynomials: $f(0,0), f(0,1), f(1,1)$ form the coefficients (i.e., control points) of F relative to the Bernstein basis.

Lemma 109

Every polynomial can be expressed in Bezier form. That is, for every polynomial $P\colon \mathbb{R} \to \mathbb{R}^2$ of degree n there exist control points $p_0, p_1, \ldots, p_n \in \mathbb{R}^2$ such that the Bézier curve defined by them matches $P|_{[0,1]}$.

Sketch of proof : Let f be the polarform of P, and let

$$p_k := f(\underbrace{0,\ldots,0}_{n-k},\underbrace{1,\ldots,1}_{k}) \quad \text{for } k = 0, 1, \ldots, n.$$

□

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 109

Every polynomial can be expressed in Bezier form. That is, for every polynomial $P\colon \mathbb{R} \to \mathbb{R}^2$ of degree n there exist control points $p_0, p_1, \ldots, p_n \in \mathbb{R}^2$ such that the Bézier curve defined by them matches $P|_{[0,1]}$.

Sketch of proof: Let f be the polarform of P, and let

$$p_k := f(\underbrace{0, \ldots, 0}_{n-k}, \underbrace{1, \ldots, 1}_{k}) \quad \text{for } k = 0, 1, \ldots, n.$$

□

- Polar forms are useful because they provide a uniform and simple means for computing values of a polynomial using a variety of representations (Bézier, B-spline, NURBS, etc.).
- For this reason, some authors prefer to introduce Bézier curves in their polar form.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Derivatives of a Bézier Curve

Lemma 110

Let $\mathcal{B}$ be a Bézier curve of degree n with $n+1$ control points $p_0, p_1, \ldots, p_n$. Its first derivative, which is sometimes called *hodograph*, is a Bézier curve of degree $n-1$,

$$\mathcal{B}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big),$$

whose n control points are given by $n(p_1 - p_0), n(p_2 - p_1), \cdots, n(p_n - p_{n-1})$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Derivatives of a Bézier Curve

Lemma 110

Let $\mathcal{B}$ be a Bézier curve of degree n with $n+1$ control points $p_0, p_1, \ldots, p_n$. Its first derivative, which is sometimes called *hodograph*, is a Bézier curve of degree $n-1$,

$$\mathcal{B}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big),$$

whose n control points are given by $n(p_1 - p_0), n(p_2 - p_1), \cdots, n(p_n - p_{n-1})$.

Proof : Since the control points are constants, computing the derivative of a Bézier curve is reduced to computing the derivatives of the Bernstein basis polynomials.

Derivatives of a Bézier Curve

Lemma 110

Let $\mathcal{B}$ be a Bézier curve of degree n with $n+1$ control points $p_0, p_1, \ldots, p_n$. Its first derivative, which is sometimes called *hodograph*, is a Bézier curve of degree $n-1$,

$$\mathcal{B}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big),$$

whose n control points are given by $n(p_1 - p_0), n(p_2 - p_1), \cdots, n(p_n - p_{n-1})$.

Proof : Since the control points are constants, computing the derivative of a Bézier curve is reduced to computing the derivatives of the Bernstein basis polynomials.

$$\mathcal{B}'(t) = \frac{d}{dt}\left(\sum_{i=0}^{n} B_{i,n}(t)p_i\right) = \sum_{i=0}^{n} B'_{i,n}(t)p_i$$

Derivatives of a Bézier Curve

Lemma 110

Let $\mathcal{B}$ be a Bézier curve of degree n with $n+1$ control points $p_0, p_1, \ldots, p_n$. Its first derivative, which is sometimes called *hodograph*, is a Bézier curve of degree $n-1$,

$$\mathcal{B}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big),$$

whose n control points are given by $n(p_1 - p_0), n(p_2 - p_1), \cdots, n(p_n - p_{n-1})$.

Proof : Since the control points are constants, computing the derivative of a Bézier curve is reduced to computing the derivatives of the Bernstein basis polynomials.

$$\mathcal{B}'(t) = \frac{d}{dt}\left(\sum_{i=0}^{n} B_{i,n}(t)p_i\right) = \sum_{i=0}^{n} B'_{i,n}(t)p_i \overset{\text{Cor. 96}}{=} n\left(\sum_{i=0}^{n}\Big(B_{i-1,n-1}(t) - B_{i,n-1}(t)\Big)p_i\right)$$

Lemma 110

Let $\mathcal{B}$ be a Bézier curve of degree n with $n+1$ control points $p_0, p_1, \ldots, p_n$. Its first derivative, which is sometimes called *hodograph*, is a Bézier curve of degree $n-1$,

$$\mathcal{B}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big),$$

whose n control points are given by $n(p_1 - p_0), n(p_2 - p_1), \cdots, n(p_n - p_{n-1})$.

Proof: Since the control points are constants, computing the derivative of a Bézier curve is reduced to computing the derivatives of the Bernstein basis polynomials.

$$\mathcal{B}'(t) = \frac{d}{dt}\left(\sum_{i=0}^{n} B_{i,n}(t)p_i\right) = \sum_{i=0}^{n} B'_{i,n}(t)p_i \overset{\text{Cor. 96}}{=} n\left(\sum_{i=0}^{n}\Big(B_{i-1,n-1}(t) - B_{i,n-1}(t)\Big)p_i\right)$$

$$= n \cdot \left(\sum_{i=1}^{n} B_{i-1,n-1}(t)p_i - \sum_{i=0}^{n-1} B_{i,n-1}(t)p_i\right)$$

Lemma 110

Let $\mathcal{B}$ be a Bézier curve of degree n with $n+1$ control points $p_0, p_1, \ldots, p_n$. Its first derivative, which is sometimes called *hodograph*, is a Bézier curve of degree $n-1$,

$$\mathcal{B}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big),$$

whose n control points are given by $n(p_1 - p_0), n(p_2 - p_1), \cdots, n(p_n - p_{n-1})$.

Proof : Since the control points are constants, computing the derivative of a Bézier curve is reduced to computing the derivatives of the Bernstein basis polynomials.

$$\mathcal{B}'(t) = \frac{d}{dt}\left(\sum_{i=0}^{n} B_{i,n}(t)p_i\right) = \sum_{i=0}^{n} B'_{i,n}(t)p_i \overset{\text{Cor. 96}}{=} n\left(\sum_{i=0}^{n}\Big(B_{i-1,n-1}(t) - B_{i,n-1}(t)\Big)p_i\right)$$

$$= n \cdot \left(\sum_{i=1}^{n} B_{i-1,n-1}(t)p_i - \sum_{i=0}^{n-1} B_{i,n-1}(t)p_i\right)$$

$$= n \cdot \left(\sum_{i=0}^{n-1} B_{i,n-1}(t)p_{i+1} - \sum_{i=0}^{n-1} B_{i,n-1}(t)p_i\right)$$

Derivatives of a Bézier Curve

Lemma 110

Let $\mathcal{B}$ be a Bézier curve of degree n with $n+1$ control points $p_0, p_1, \ldots, p_n$. Its first derivative, which is sometimes called *hodograph*, is a Bézier curve of degree $n-1$,

$$\mathcal{B}'(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big),$$

whose n control points are given by $n(p_1 - p_0), n(p_2 - p_1), \cdots, n(p_n - p_{n-1})$.

Proof : Since the control points are constants, computing the derivative of a Bézier curve is reduced to computing the derivatives of the Bernstein basis polynomials.

$$\mathcal{B}'(t) = \frac{d}{dt}\left(\sum_{i=0}^{n} B_{i,n}(t)p_i\right) = \sum_{i=0}^{n} B'_{i,n}(t)p_i \overset{\text{Cor. 96}}{=} n\left(\sum_{i=0}^{n}\big(B_{i-1,n-1}(t) - B_{i,n-1}(t)\big)p_i\right)$$

$$= n \cdot \left(\sum_{i=1}^{n} B_{i-1,n-1}(t)p_i - \sum_{i=0}^{n-1} B_{i,n-1}(t)p_i\right)$$

$$= n \cdot \left(\sum_{i=0}^{n-1} B_{i,n-1}(t)p_{i+1} - \sum_{i=0}^{n-1} B_{i,n-1}(t)p_i\right) = \sum_{i=0}^{n-1} B_{i,n-1}(t)\big(n(p_{i+1} - p_i)\big)$$

Lemma 111

A Bézier curve is tangent to the control polygon at the endpoints.

Lemma 111

A Bézier curve is tangent to the control polygon at the endpoints.

Proof : This is readily proved by computing $\mathcal{B}'(0)$ and $\mathcal{B}'(1)$. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 111

A Bézier curve is tangent to the control polygon at the endpoints.

Proof : This is readily proved by computing $\mathcal{B}'(0)$ and $\mathcal{B}'(1)$. □

- Hence, joining two Bézier curves in a G^1-continuous way is easy.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 111

A Bézier curve is tangent to the control polygon at the endpoints.

Proof: This is readily proved by computing $\mathcal{B}'(0)$ and $\mathcal{B}'(1)$. □

- Hence, joining two Bézier curves in a G^1-continuous way is easy.
- Let $p_0, p_1, \ldots, p_n$ and $p_0^\star, p_1^\star, \ldots, p_m^\star$ be the control points of two Bézier curves $\mathcal{B}$ and $\mathcal{B}^\star$. In order to achieve C^1-continuity, we need (in addition to $p_n = p_0^\star$)

$$\mathcal{B}'(1) = (\mathcal{B}^\star)'(0) \qquad \text{i.e.,} \quad n(p_n - p_{n-1}) = m(p_1^\star - p_0^\star).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 111

A Bézier curve is tangent to the control polygon at the endpoints.

Proof: This is readily proved by computing $\mathcal{B}'(0)$ and $\mathcal{B}'(1)$. □

- Hence, joining two Bézier curves in a G^1-continuous way is easy.
- Let $p_0, p_1, \ldots, p_n$ and $p_0^\star, p_1^\star, \ldots, p_m^\star$ be the control points of two Bézier curves $\mathcal{B}$ and $\mathcal{B}^\star$. In order to achieve C^1-continuity, we need (in addition to $p_n = p_0^\star$)

$$\mathcal{B}'(1) = (\mathcal{B}^\star)'(0) \qquad \text{i.e.,} \;\; n(p_n - p_{n-1}) = m(p_1^\star - p_0^\star).$$

- This has an interesting consequence for closed Bézier curves with $p_0 = \mathcal{B}(0) = \mathcal{B}(1) = p_n$:

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 111

A Bézier curve is tangent to the control polygon at the endpoints.

Proof : This is readily proved by computing $\mathcal{B}'(0)$ and $\mathcal{B}'(1)$. □

- Hence, joining two Bézier curves in a G^1-continuous way is easy.
- Let $p_0, p_1, \ldots, p_n$ and $p_0^\star, p_1^\star, \ldots, p_m^\star$ be the control points of two Bézier curves $\mathcal{B}$ and $\mathcal{B}^\star$. In order to achieve C^1-continuity, we need (in addition to $p_n = p_0^\star$)

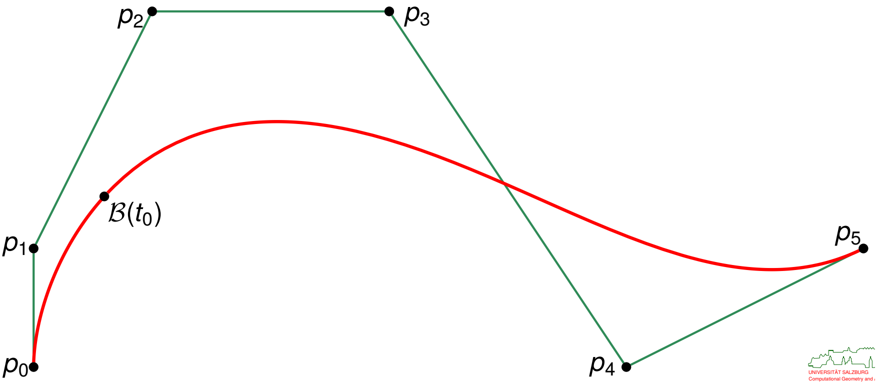
$$\mathcal{B}'(1) = (\mathcal{B}^\star)'(0) \qquad \text{i.e., } n(p_n - p_{n-1}) = m(p_1^\star - p_0^\star).$$

- This has an interesting consequence for closed Bézier curves with $p_0 = \mathcal{B}(0) = \mathcal{B}(1) = p_n$:
  - We get G^1-continuity at p_0 if p_0, p_1, p_{n-1} are collinear.
  - We get C^1-continuity at p_0 if $p_1 - p_0 = p_n - p_{n-1}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

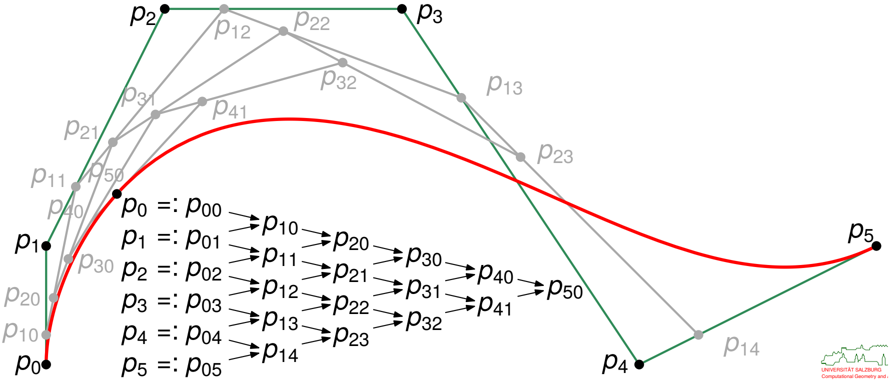
Subdivision of a Bézier Curve

- One can subdivide a Bézier curve $\mathcal{B}$ of degree n into two curves, at a point $\mathcal{B}(t_0)$ for a given parameter t_0, such that the newly obtained Bézier curves $\mathcal{B}_1$ and $\mathcal{B}_2$ have their own set of control points and are of degree n each:

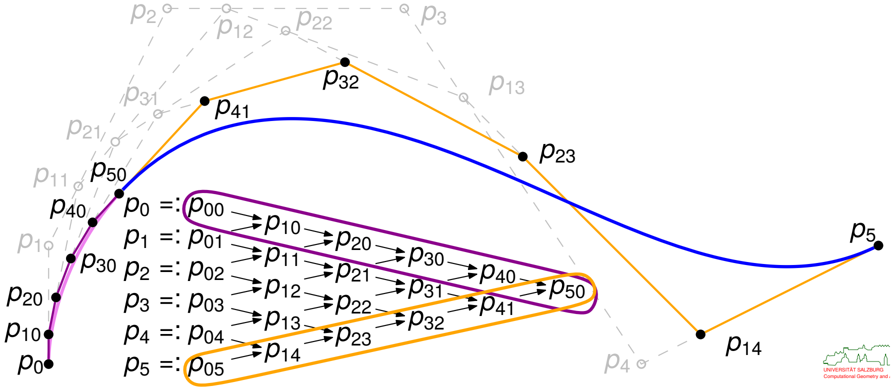


Subdivision of a Bézier Curve

- One can subdivide a Bézier curve $\mathcal{B}$ of degree n into two curves, at a point $\mathcal{B}(t_0)$ for a given parameter t_0, such that the newly obtained Bézier curves $\mathcal{B}_1$ and $\mathcal{B}_2$ have their own set of control points and are of degree n each:
  - First, we use de Casteljau's algorithm to compute $\mathcal{B}(t_0)$.

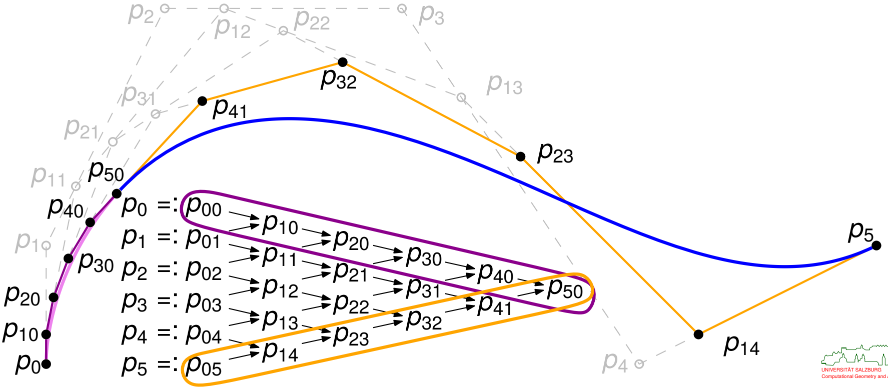


- One can subdivide a Bézier curve $\mathcal{B}$ of degree n into two curves, at a point $\mathcal{B}(t_0)$ for a given parameter t_0, such that the newly obtained Bézier curves $\mathcal{B}_1$ and $\mathcal{B}_2$ have their own set of control points and are of degree n each:
  - First, we use de Casteljau's algorithm to compute $\mathcal{B}(t_0)$.
  - The de Casteljau net can then be used to generate the new control polygons for $\mathcal{B}_1$ and $\mathcal{B}_2$.



Subdivision of a Bézier Curve

- One can subdivide a Bézier curve $\mathcal{B}$ of degree n into two curves, at a point $\mathcal{B}(t_0)$ for a given parameter t_0, such that the newly obtained Bézier curves $\mathcal{B}_1$ and $\mathcal{B}_2$ have their own set of control points and are of degree n each:
  - First, we use de Casteljau's algorithm to compute $\mathcal{B}(t_0)$.
  - The de Casteljau net can then be used to generate the new control polygons for $\mathcal{B}_1$ and $\mathcal{B}_2$.
  - Note that $\mathcal{B}_1$ and $\mathcal{B}_2$ join in a G^1-continuous way.



Lemma 112

Let $p_0, p_1, \dots, p_n$ be the control points of the Bézier curve $\mathcal{B}$, and let $p_{i,j}$ denote the control points obtained by de Casteljau's algorithm for some $t_0 \in]0,1[$. We define new control points as follows:

$$p_i^\star := p_{i,0} \qquad \text{for } i = 0, 1, \dots, n$$
$$p_i^{\star\star} := p_{n-i,i} \qquad \text{for } i = 0, 1, \dots, n$$

Let $\mathcal{B}^\star$ ($\mathcal{B}^{\star\star}$, resp.) denote the Bézier curve defined by $p_0^\star, p_1^\star, \dots, p_n^\star$ ($p_0^{\star\star}, p_1^{\star\star}, \dots, p_n^{\star\star}$, resp.). Then $\mathcal{B}^\star$ and $\mathcal{B}^{\star\star}$ join in a tangent-continuous way at point $p_n^\star = p_0^{\star\star}$, and we have

$$\mathcal{B}^\star = \mathcal{B}|_{[0,t_0]} \quad \text{and} \quad \mathcal{B}^{\star\star} = \mathcal{B}|_{[t_0,1]}.$$

Subdivision of a Bézier Curve

Lemma 112

Let $p_0, p_1, \ldots, p_n$ be the control points of the Bézier curve $\mathcal{B}$, and let $p_{i,j}$ denote the control points obtained by de Casteljau's algorithm for some $t_0 \in]0,1[$. We define new control points as follows:

$$p_i^\star := p_{i,0} \qquad \text{for } i = 0, 1, \ldots, n$$
$$p_i^{\star\star} := p_{n-i,i} \qquad \text{for } i = 0, 1, \ldots, n$$

Let $\mathcal{B}^\star$ ($\mathcal{B}^{\star\star}$, resp.) denote the Bézier curve defined by $p_0^\star, p_1^\star, \ldots, p_n^\star$ ($p_0^{\star\star}, p_1^{\star\star}, \ldots, p_n^{\star\star}$, resp.). Then $\mathcal{B}^\star$ and $\mathcal{B}^{\star\star}$ join in a tangent-continuous way at point $p_n^\star = p_0^{\star\star}$, and we have

$$\mathcal{B}^\star = \mathcal{B}|_{[0,t_0]} \quad \text{and} \quad \mathcal{B}^{\star\star} = \mathcal{B}|_{[t_0,1]}.$$

- Note: With every subdivision the control polygons get closer to the Bézier curve. And the approximation is quite fast: For k (uniform recursive) subdivision steps, the maximum distance ε between the resulting control polygon and the curve is

$$\varepsilon < \frac{c}{2^k} \qquad \text{for some positive constant } c.$$

Degree Elevation of a Bézier Curve

- An increase of the number of control points of a Bézier curve increases the flexibility in designing shapes.
- The key goal is to preserve the shape of the curve. (Recall that Bézier curves change globally if one control point is relocated!)
- Of course, adding one control point means increasing the degree of a Bézier curve by one.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

- An increase of the number of control points of a Bézier curve increases the flexibility in designing shapes.
- The key goal is to preserve the shape of the curve. (Recall that Bézier curves change globally if one control point is relocated!)
- Of course, adding one control point means increasing the degree of a Bézier curve by one.
- Let $p_0, p_1, \ldots, p_n$ be the old control points, and $p_0^\star, p_1^\star, \ldots, p_n^\star, p_{n+1}^\star$ be the new control points, and denote the Bézier curves defined by them by $\mathcal{B}$ and $\mathcal{B}^\star$.
- How can we guarantee $\mathcal{B}(t) = \mathcal{B}^\star(t)$ for all $t \in [0, 1]$?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

- An increase of the number of control points of a Bézier curve increases the flexibility in designing shapes.
- The key goal is to preserve the shape of the curve. (Recall that Bézier curves change globally if one control point is relocated!)
- Of course, adding one control point means increasing the degree of a Bézier curve by one.
- Let $p_0, p_1, \dots, p_n$ be the old control points, and $p_0^\star, p_1^\star, \dots, p_n^\star, p_{n+1}^\star$ be the new control points, and denote the Bézier curves defined by them by $\mathcal{B}$ and $\mathcal{B}^\star$.
- How can we guarantee $\mathcal{B}(t) = \mathcal{B}^\star(t)$ for all $t \in [0, 1]$?
- Obviously, we will need

$$p_0 = p_0^\star \quad \text{and} \quad p_n = p_{n+1}^\star$$

  in order to ensure that at least the start and end points of $\mathcal{B}$ and $\mathcal{B}^\star$ match.
- In the sequel, we will find it convenient to extend the index range of the control points of $\mathcal{B}$ and introduce (arbitrary) points p_{-1} and p_{n+1}. (Both points will be multiplied with factors that equal zero, anyway.)

Degree Elevation of a Bézier Curve

- Standard equalities:

$$\binom{n+1}{i}(1-t)\cdot B_{i,n}(t) = \binom{n+1}{i}(1-t)\binom{n}{i}t^i(1-t)^{n-i}$$

Degree Elevation of a Bézier Curve

- Standard equalities:

$$\binom{n+1}{i}(1-t)\cdot B_{i,n}(t) = \binom{n+1}{i}(1-t)\binom{n}{i}t^i(1-t)^{n-i}$$
$$= \binom{n}{i}\binom{n+1}{i}t^i(1-t)^{n+1-i} = \binom{n}{i}B_{i,n+1}(t)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

- Standard equalities:

$$\binom{n+1}{i}(1-t)\cdot B_{i,n}(t) = \binom{n+1}{i}(1-t)\binom{n}{i}t^i(1-t)^{n-i}$$

$$= \binom{n}{i}\binom{n+1}{i}t^i(1-t)^{n+1-i} = \binom{n}{i}B_{i,n+1}(t)$$

and

$$\binom{n+1}{i+1}t\cdot B_{i,n}(t) = \binom{n+1}{i+1}t\binom{n}{i}t^i(1-t)^{n-i}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

- Standard equalities:

$$\binom{n+1}{i}(1-t)\cdot B_{i,n}(t) = \binom{n+1}{i}(1-t)\binom{n}{i}t^i(1-t)^{n-i}$$
$$= \binom{n}{i}\binom{n+1}{i}t^i(1-t)^{n+1-i} = \binom{n}{i}B_{i,n+1}(t)$$

and

$$\binom{n+1}{i+1}t\cdot B_{i,n}(t) = \binom{n+1}{i+1}t\binom{n}{i}t^i(1-t)^{n-i}$$
$$= \binom{n}{i}\binom{n+1}{i+1}t^{i+1}(1-t)^{n-i} = \binom{n}{i}B_{i+1,n+1}(t)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

- Standard equalities:

$$\binom{n+1}{i}(1-t)\cdot B_{i,n}(t) = \binom{n+1}{i}(1-t)\binom{n}{i}t^i(1-t)^{n-i}$$
$$= \binom{n}{i}\binom{n+1}{i}t^i(1-t)^{n+1-i} = \binom{n}{i}B_{i,n+1}(t)$$

and

$$\binom{n+1}{i+1}t\cdot B_{i,n}(t) = \binom{n+1}{i+1}t\binom{n}{i}t^i(1-t)^{n-i}$$
$$= \binom{n}{i}\binom{n+1}{i+1}t^{i+1}(1-t)^{n-i} = \binom{n}{i}B_{i+1,n+1}(t)$$

- Hence,

$$(1-t)\cdot B_{i,n}(t) = \frac{n+1-i}{n+1}B_{i,n+1}(t) \quad \text{and} \quad t\cdot B_{i,n}(t) = \frac{i+1}{n+1}B_{i+1,n+1}(t).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

$$\mathcal{B}(t) = \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i$$

Degree Elevation of a Bézier Curve

$$\mathcal{B}(t) = \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i$$

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

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

$$\begin{aligned}\mathcal{B}(t) &= \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i \\ &= (1-t)\sum_{i=0}^{n} B_{i,n}(t)p_i + t\sum_{i=0}^{n} B_{i,n}(t)p_i = \sum_{i=0}^{n}(1-t)\cdot B_{i,n}(t)p_i + \sum_{i=0}^{n} t\cdot B_{i,n}(t)p_i\end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

$$\begin{aligned}
\mathcal{B}(t) &= \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i \\
&= (1-t)\sum_{i=0}^{n} B_{i,n}(t)p_i + t\sum_{i=0}^{n} B_{i,n}(t)p_i = \sum_{i=0}^{n}(1-t)\cdot B_{i,n}(t)p_i + \sum_{i=0}^{n} t\cdot B_{i,n}(t)p_i \\
&= \sum_{i=0}^{n} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=0}^{n} \frac{i+1}{n+1} B_{i+1,n+1}(t)p_i
\end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

$$\begin{aligned}
\mathcal{B}(t) &= \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i \\
&= (1-t)\sum_{i=0}^{n} B_{i,n}(t)p_i + t\sum_{i=0}^{n} B_{i,n}(t)p_i = \sum_{i=0}^{n}(1-t)\cdot B_{i,n}(t)p_i + \sum_{i=0}^{n} t\cdot B_{i,n}(t)p_i \\
&= \sum_{i=0}^{n} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=0}^{n} \frac{i+1}{n+1} B_{i+1,n+1}(t)p_i \\
&= \sum_{i=0}^{n+1} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=-1}^{n} \frac{i+1}{n+1} B_{i+1,n+1}(t)p_i
\end{aligned}$$

Degree Elevation of a Bézier Curve

$$\begin{aligned}
\mathcal{B}(t) &= \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i \\
&= (1-t)\sum_{i=0}^{n} B_{i,n}(t)p_i + t\sum_{i=0}^{n} B_{i,n}(t)p_i = \sum_{i=0}^{n}(1-t)\cdot B_{i,n}(t)p_i + \sum_{i=0}^{n} t\cdot B_{i,n}(t)p_i \\
&= \sum_{i=0}^{n}\frac{n+1-i}{n+1}B_{i,n+1}(t)p_i + \sum_{i=0}^{n}\frac{i+1}{n+1}B_{i+1,n+1}(t)p_i \\
&= \sum_{i=0}^{n+1}\frac{n+1-i}{n+1}B_{i,n+1}(t)p_i + \sum_{i=-1}^{n}\frac{i+1}{n+1}B_{i+1,n+1}(t)p_i \\
&= \sum_{i=0}^{n+1}\frac{n+1-i}{n+1}B_{i,n+1}(t)p_i + \sum_{i=0}^{n+1}\frac{i}{n+1}B_{i,n+1}(t)p_{i-1}
\end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

$$
\begin{aligned}
\mathcal{B}(t) &= \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i \\
&= (1-t)\sum_{i=0}^{n} B_{i,n}(t)p_i + t\sum_{i=0}^{n} B_{i,n}(t)p_i = \sum_{i=0}^{n}(1-t)\cdot B_{i,n}(t)p_i + \sum_{i=0}^{n} t\cdot B_{i,n}(t)p_i \\
&= \sum_{i=0}^{n} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=0}^{n} \frac{i+1}{n+1} B_{i+1,n+1}(t)p_i \\
&= \sum_{i=0}^{n+1} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=-1}^{n} \frac{i+1}{n+1} B_{i+1,n+1}(t)p_i \\
&= \sum_{i=0}^{n+1} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=0}^{n+1} \frac{i}{n+1} B_{i,n+1}(t)p_{i-1} \\
&= \sum_{i=0}^{n+1} B_{i,n+1}(t)\left(\frac{i}{n+1}p_{i-1} + \frac{n+1-i}{n+1}p_i\right)
\end{aligned}
$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

$$
\begin{aligned}
\mathcal{B}(t) &= \sum_{i=0}^{n} B_{i,n}(t)p_i = ((1-t)+t)\sum_{i=0}^{n} B_{i,n}(t)p_i \\
&= (1-t)\sum_{i=0}^{n} B_{i,n}(t)p_i + t\sum_{i=0}^{n} B_{i,n}(t)p_i = \sum_{i=0}^{n}(1-t)\cdot B_{i,n}(t)p_i + \sum_{i=0}^{n} t\cdot B_{i,n}(t)p_i \\
&= \sum_{i=0}^{n} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=0}^{n} \frac{i+1}{n+1} B_{i+1,n+1}(t)p_i \\
&= \sum_{i=0}^{n+1} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=-1}^{n} \frac{i+1}{n+1} B_{i+1,n+1}(t)p_i \\
&= \sum_{i=0}^{n+1} \frac{n+1-i}{n+1} B_{i,n+1}(t)p_i + \sum_{i=0}^{n+1} \frac{i}{n+1} B_{i,n+1}(t)p_{i-1} \\
&= \sum_{i=0}^{n+1} B_{i,n+1}(t)\left(\frac{i}{n+1}p_{i-1} + \frac{n+1-i}{n+1}p_i\right) = \sum_{i=0}^{n+1} B_{i,n+1}(t)p_i^\star =: \mathcal{B}^\star(t)
\end{aligned}
$$

with

$$
p_i^\star := \frac{i}{n+1}p_{i-1} + \frac{n+1-i}{n+1}p_i, \qquad i = 0, \ldots, n+1.
$$

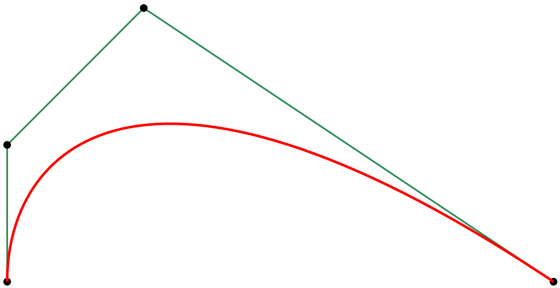
Lemma 113

Let $p_0, p_1, \ldots, p_n$ be the control points of the degree-n Bézier curve $\mathcal{B}$. If we use

$$p_i^\star := \left(\frac{i}{n+1}\right) p_{i-1} + \left(1 - \frac{i}{n+1}\right) p_i \qquad \text{for } \; i = 0, 1, \cdots, n+1$$

as control points for the Bézier curve $\mathcal{B}^\star$ of degree $n+1$, then

$$\mathcal{B}(t) = \mathcal{B}^\star(t) \qquad \text{for all } \; t \in [0, 1].$$



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Degree Elevation of a Bézier Curve

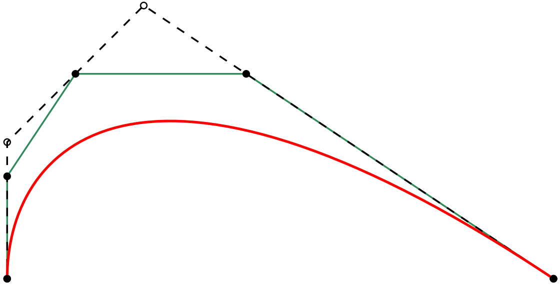
Lemma 113

Let $p_0, p_1, \ldots, p_n$ be the control points of the degree-n Bézier curve $\mathcal{B}$. If we use

$$p_i^\star := \left(\frac{i}{n+1}\right) p_{i-1} + \left(1 - \frac{i}{n+1}\right) p_i \qquad \text{for } \; i = 0, 1, \cdots, n+1$$

as control points for the Bézier curve $\mathcal{B}^\star$ of degree $n+1$, then

$$\mathcal{B}(t) = \mathcal{B}^\star(t) \qquad \text{for all } \; t \in [0, 1].$$



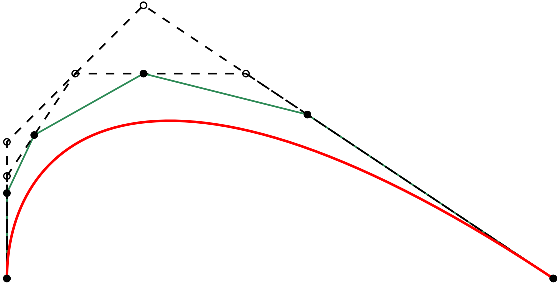
Lemma 113

Let $p_0, p_1, \dots, p_n$ be the control points of the degree-n Bézier curve $\mathcal{B}$. If we use

$$p_i^\star := \left(\frac{i}{n+1}\right) p_{i-1} + \left(1 - \frac{i}{n+1}\right) p_i \qquad \text{for } \ i = 0, 1, \cdots, n+1$$

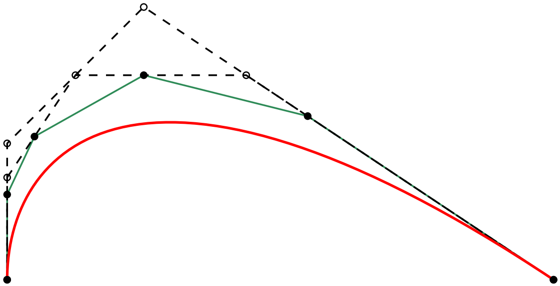
as control points for the Bézier curve $\mathcal{B}^\star$ of degree $n+1$, then

$$\mathcal{B}(t) = \mathcal{B}^\star(t) \qquad \text{for all } \ t \in [0, 1].$$



Degree Elevation of a Bézier Curve

- Note that all newly created control points lie on the edges of the previous control polygon.
- Effectively, the corners of the previous control polygon are cut off.
- Degree elevation can be used repeatedly, e.g., in order to arrive at the same degrees for two Bézier curves that join.
- As the degree keeps increasing, the control polyon approaches the Bézier curve and has it as a limiting position.



Bernstein Basis Polynomials and Matrix Representation

- Consider Bernstein basis polynomials of degree three:

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

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials and Matrix Representation

- Consider Bernstein basis polynomials of degree three:

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

- By applying the Binomial Theorem 10, we get $B_{0,3}(t) = 1 - 3t + 3t^2 - t^3$.

$$\begin{pmatrix} B_{03}(t) \\ B_{13}(t) \\ B_{23}(t) \\ B_{33}(t) \end{pmatrix} = \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials and Matrix Representation

- Consider Bernstein basis polynomials of degree three:

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

- By applying the Binomial Theorem 10, we get $B_{0,3}(t) = 1 - 3t + 3t^2 - t^3$.

$$\begin{pmatrix} B_{03}(t) \\ B_{13}(t) \\ B_{23}(t) \\ B_{33}(t) \end{pmatrix} = \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

- We get

$$\mathbf{B}^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & 0 & \frac{1}{3} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \text{for } \mathbf{B} := \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Basis Polynomials and Matrix Representation

- Consider Bernstein basis polynomials of degree three:

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

- By applying the Binomial Theorem 10, we get $B_{0,3}(t) = 1 - 3t + 3t^2 - t^3$.

$$\begin{pmatrix} B_{03}(t) \\ B_{13}(t) \\ B_{23}(t) \\ B_{33}(t) \end{pmatrix} = \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

- We get

$$\mathbf{B}^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & 0 & \frac{1}{3} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \text{for } \mathbf{B} := \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- Hence, $\mathbf{B}^{-1}$ allows a basis conversion from power basis to Bernstein basis:

$$\begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & 0 & \frac{1}{3} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} B_{03}(t) \\ B_{13}(t) \\ B_{23}(t) \\ B_{33}(t) \end{pmatrix}$$

Matrix Representation of Bézier Curve

- Since $\mathcal{B}(t) = \sum_{i=0}^{3} B_{i,3}(t)p_i$, we obtain

$$\mathcal{B}(t) = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

as the matrix representation of a cubic Bézier curve.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Matrix Representation of Bézier Curve

- Since $\mathcal{B}(t) = \sum_{i=0}^{3} B_{i,3}(t)p_i$, we obtain

$$\mathcal{B}(t) = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

  as the matrix representation of a cubic Bézier curve.
- This approach can be generalized to representing a degree-n Bézier curve by an $(n+1) \times (n+1)$ matrix.

Application of Matrix Representation

- The matrix representation gives a simple way to prove Lemma 106: For a linear transformation with matrix **A**, we get

$$\mathbf{A}\cdot\mathcal{B}(t) = \mathbf{A}\cdot\left(\begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix}\cdot\mathbf{B}\cdot\begin{pmatrix}1\\ t\\ t^2\\ t^3\end{pmatrix}\right) = \left(\mathbf{A}\cdot\begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix}\right)\cdot\mathbf{B}\cdot\begin{pmatrix}1\\ t\\ t^2\\ t^3\end{pmatrix}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Application of Matrix Representation

- The matrix representation gives a simple way to prove Lemma 106: For a linear transformation with matrix **A**, we get

$$\mathbf{A}\cdot\mathcal{B}(t) = \mathbf{A}\cdot\left(\begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix}\cdot\mathbf{B}\cdot\begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}\right) = \left(\mathbf{A}\cdot\begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix}\right)\cdot\mathbf{B}\cdot\begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}.$$

- Derivatives are obtained in a similar way:

$$\mathcal{B}'(t) = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix}\cdot\mathbf{B}\cdot\begin{pmatrix} 0 \\ 1 \\ 2t \\ 3t^2 \end{pmatrix}.$$

Application of Matrix Representation

- Suppose that we want to subdivide $\mathcal{B}(t)$ for $t := {}^1\!/_2$.

Application of Matrix Representation

- Suppose that we want to subdivide $\mathcal{B}(t)$ for $t := 1/2$.
- This can be done by defining a new curve $\mathcal{B}_1(t)$ that is equal to $\mathcal{B}(t/2)$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Application of Matrix Representation

- Suppose that we want to subdivide $\mathcal{B}(t)$ for $t := {}^1\!/_2$.
- This can be done by defining a new curve $\mathcal{B}_1(t)$ that is equal to $\mathcal{B}({}^t\!/_2)$, and we get

$$\mathcal{B}_1(t) = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 \\ {}^t\!/_2 \\ ({}^t\!/_2)^2 \\ ({}^t\!/_2)^3 \end{pmatrix}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Application of Matrix Representation

- Suppose that we want to subdivide $\mathcal{B}(t)$ for $t := 1/2$.
- This can be done by defining a new curve $\mathcal{B}_1(t)$ that is equal to $\mathcal{B}(t/2)$, and we get

$$\mathcal{B}_1(t) = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 \\ t/2 \\ (t/2)^2 \\ (t/2)^3 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/8 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

Application of Matrix Representation

- Suppose that we want to subdivide $\mathcal{B}(t)$ for $t := {}^1\!/_2$.
- This can be done by defining a new curve $\mathcal{B}_1(t)$ that is equal to $\mathcal{B}({}^t\!/_2)$, and we get

$$\mathcal{B}_1(t) = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 \\ {}^t\!/_2 \\ ({}^t\!/_2)^2 \\ ({}^t\!/_2)^3 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & {}^1\!/_2 & 0 & 0 \\ 0 & 0 & {}^1\!/_4 & 0 \\ 0 & 0 & 0 & {}^1\!/_8 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & {}^1\!/_2 & 0 & 0 \\ 0 & 0 & {}^1\!/_4 & 0 \\ 0 & 0 & 0 & {}^1\!/_8 \end{pmatrix} \cdot \mathbf{B}^{-1} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

Application of Matrix Representation

- Suppose that we want to subdivide $\mathcal{B}(t)$ for $t := 1/2$.
- This can be done by defining a new curve $\mathcal{B}_1(t)$ that is equal to $\mathcal{B}(t/2)$, and we get

$$\mathcal{B}_1(t) = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 \\ t/2 \\ (t/2)^2 \\ (t/2)^3 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/8 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/8 \end{pmatrix} \cdot \mathbf{B}^{-1} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1/2 & 1/4 & 1/8 \\ 0 & 1/2 & 1/2 & 3/8 \\ 0 & 0 & 1/4 & 3/8 \\ 0 & 0 & 0 & 1/8 \end{pmatrix} \cdot \mathbf{B} \cdot \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}.$$

Application of Matrix Representation

- We conclude that the new control points of $\mathcal{B}_1(t)$ are given as follows:

$$\begin{pmatrix} p_0^* & p_1^* & p_2^* & p_3^* \end{pmatrix} = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1/2 & 1/4 & 1/8 \\ 0 & 1/2 & 1/2 & 3/8 \\ 0 & 0 & 1/4 & 3/8 \\ 0 & 0 & 0 & 1/8 \end{pmatrix}$$

Application of Matrix Representation

- We conclude that the new control points of $\mathcal{B}_1(t)$ are given as follows:

$$\begin{pmatrix} p_0^* & p_1^* & p_2^* & p_3^* \end{pmatrix} = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1/2 & 1/4 & 1/8 \\ 0 & 1/2 & 1/2 & 3/8 \\ 0 & 0 & 1/4 & 3/8 \\ 0 & 0 & 0 & 1/8 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & \frac{1}{2}p_0 + \frac{1}{2}p_1 & \frac{1}{4}p_0 + \frac{1}{2}p_1 + \frac{1}{4}p_2 & \frac{1}{8}p_0 + \frac{3}{8}p_1 + \frac{3}{8}p_2 + \frac{1}{8}p_3 \end{pmatrix}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Application of Matrix Representation

- We conclude that the new control points of $\mathcal{B}_1(t)$ are given as follows:

$$\begin{pmatrix} p_0^* & p_1^* & p_2^* & p_3^* \end{pmatrix} = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1/2 & 1/4 & 1/8 \\ 0 & 1/2 & 1/2 & 3/8 \\ 0 & 0 & 1/4 & 3/8 \\ 0 & 0 & 0 & 1/8 \end{pmatrix}$$

$$= \begin{pmatrix} p_0 & \frac{1}{2}p_0 + \frac{1}{2}p_1 & \frac{1}{4}p_0 + \frac{1}{2}p_1 + \frac{1}{4}p_2 & \frac{1}{8}p_0 + \frac{3}{8}p_1 + \frac{3}{8}p_2 + \frac{1}{8}p_3 \end{pmatrix}$$

- Similarly, the control points for the second half of the curve are obtained by studying $\mathcal{B}(\frac{1}{2}(1+t))$, yielding
$$\begin{pmatrix} \frac{1}{8}p_0 + \frac{3}{8}p_1 + \frac{3}{8}p_2 + \frac{1}{8}p_3 & \frac{1}{4}p_1 + \frac{1}{2}p_2 + \frac{1}{4}p_3 & \frac{1}{2}p_2 + \frac{1}{2}p_3 & p_3 \end{pmatrix}.$$

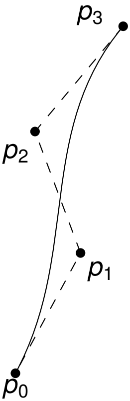
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

From Bézier Curve to Bézier Surface

- Consider a cubic Bézier curve with control points p_0, p_1, p_2, p_3:

$$\mathcal{B}(u) := B_{0,3}(u)p_0 + B_{1,3}(u)p_1 + B_{2,3}(u)p_2 + B_{3,3}(u)p_3.$$



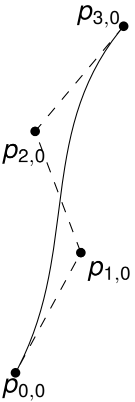
From Bézier Curve to Bézier Surface

- Consider a cubic Bézier curve with control points p_0, p_1, p_2, p_3:

$$\mathcal{B}(u) := B_{0,3}(u)p_0 + B_{1,3}(u)p_1 + B_{2,3}(u)p_2 + B_{3,3}(u)p_3.$$

- Rename control points as $p_{0,0}, p_{1,0}, p_{2,0}, p_{3,0}$:

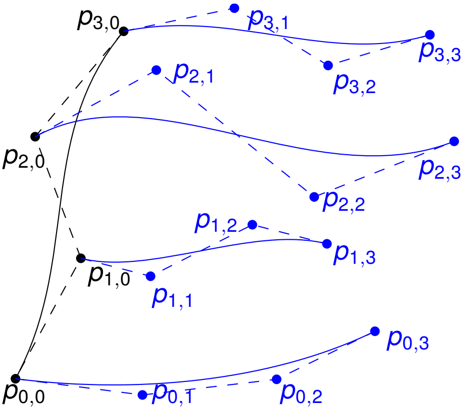
$$\mathcal{B}(u) = B_{0,3}(u)p_{0,0} + B_{1,3}(u)p_{1,0} + B_{2,3}(u)p_{2,0} + B_{3,3}(u)p_{3,0}.$$



From Bézier Curve to Bézier Surface

- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \leq i, j \leq 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$



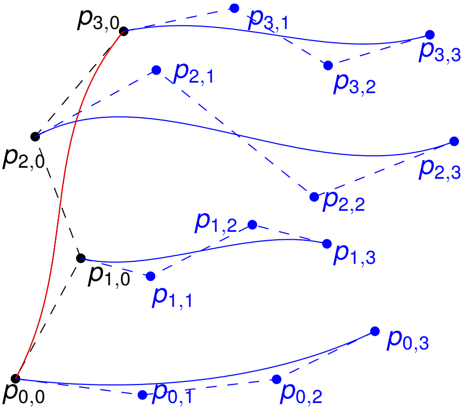
From Bézier Curve to Bézier Surface

- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \leq i, j \leq 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$

- For increasing values of $0 \leq v \leq 1$, consider

$$\mathcal{S}(u, v) := B_{0,3}(u)P_0(v) + B_{1,3}(u)P_1(v) + B_{2,3}(u)P_2(v) + B_{3,3}(u)P_3(v).$$



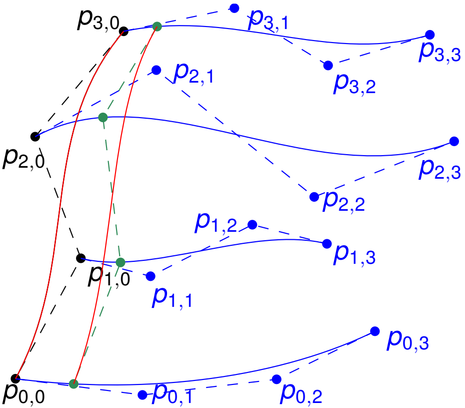
From Bézier Curve to Bézier Surface

- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \leq i, j \leq 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$

- For increasing values of $0 \leq v \leq 1$, consider

$$\mathcal{S}(u, v) := B_{0,3}(u)P_0(v) + B_{1,3}(u)P_1(v) + B_{2,3}(u)P_2(v) + B_{3,3}(u)P_3(v).$$



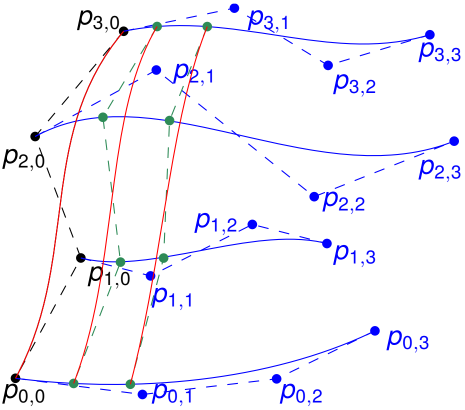
From Bézier Curve to Bézier Surface

- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \leq i, j \leq 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$

- For increasing values of $0 \leq v \leq 1$, consider

$$\mathcal{S}(u, v) := B_{0,3}(u)P_0(v) + B_{1,3}(u)P_1(v) + B_{2,3}(u)P_2(v) + B_{3,3}(u)P_3(v).$$



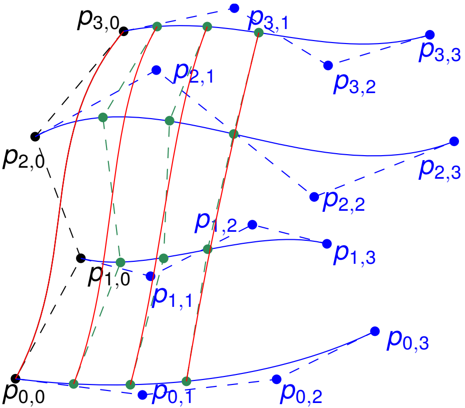
From Bézier Curve to Bézier Surface

- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \leq i, j \leq 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$

- For increasing values of $0 \leq v \leq 1$, consider

$$\mathcal{S}(u, v) := B_{0,3}(u)P_0(v) + B_{1,3}(u)P_1(v) + B_{2,3}(u)P_2(v) + B_{3,3}(u)P_3(v).$$



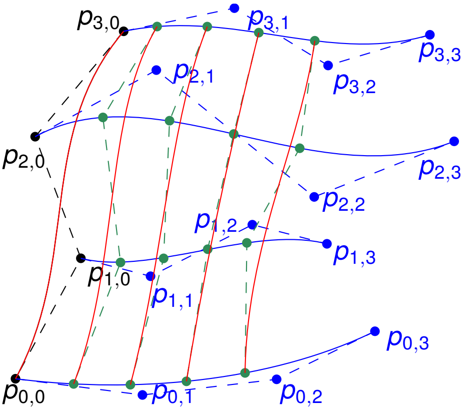
From Bézier Curve to Bézier Surface

- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \le i, j \le 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$

- For increasing values of $0 \le v \le 1$, consider

$$\mathcal{S}(u, v) := B_{0,3}(u)P_0(v) + B_{1,3}(u)P_1(v) + B_{2,3}(u)P_2(v) + B_{3,3}(u)P_3(v).$$

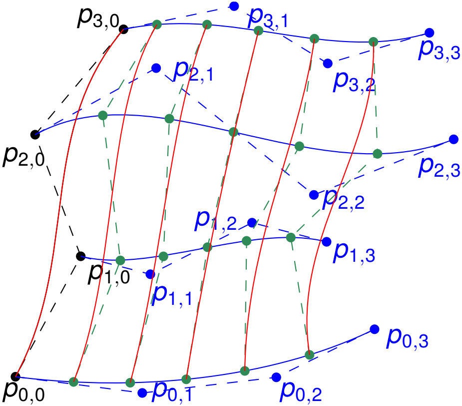


- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \leq i, j \leq 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$

- For increasing values of $0 \leq v \leq 1$, consider

$$\mathcal{S}(u, v) := B_{0,3}(u)P_0(v) + B_{1,3}(u)P_1(v) + B_{2,3}(u)P_2(v) + B_{3,3}(u)P_3(v).$$



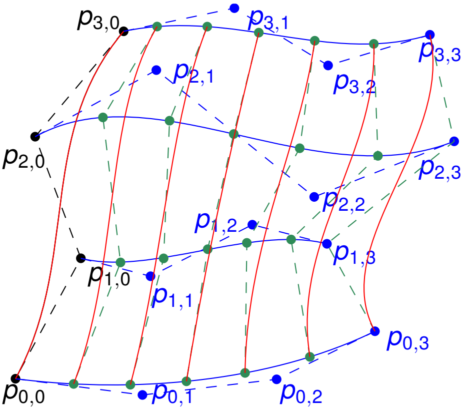
From Bézier Curve to Bézier Surface

- Add four Bézier curves $P_i(v)$ with control points $p_{i,j}$ for $0 \le i, j \le 3$. We get

$$P_i(v) = B_{0,3}(v)p_{i,0} + B_{1,3}(v)p_{i,1} + B_{2,3}(v)p_{i,2} + B_{3,3}(v)p_{i,3} \quad \text{and} \quad P_i(0) = p_{i,0} = p_i.$$

- For increasing values of $0 \le v \le 1$, consider

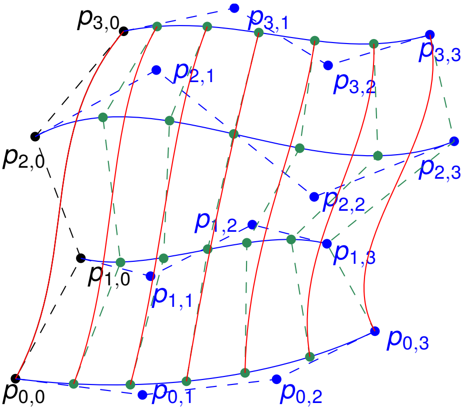
$$\mathcal{S}(u, v) := B_{0,3}(u)P_0(v) + B_{1,3}(u)P_1(v) + B_{2,3}(u)P_2(v) + B_{3,3}(u)P_3(v).$$



From Bézier Curve to Bézier Surface

- We get

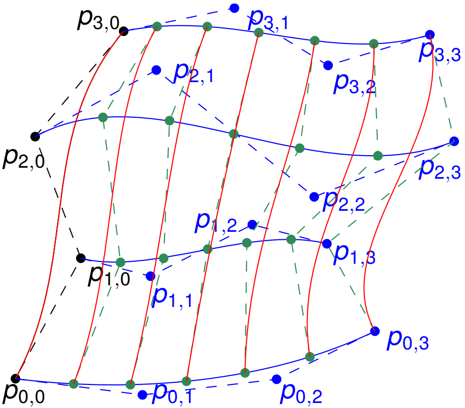
$$\mathcal{S}(u, v) = \sum_{i=0}^{3} B_{i,3}(u) P_i(v)$$



From Bézier Curve to Bézier Surface

- We get

$$\mathcal{S}(u,v) = \sum_{i=0}^{3} B_{i,3}(u) P_i(v) = \sum_{i=0}^{3} B_{i,3}(u) \sum_{j=0}^{3} B_{j,3}(v) p_{i,j} = \sum_{i=0}^{3} \sum_{j=0}^{3} B_{i,3}(u) B_{j,3}(v) p_{i,j}.$$



Definition 114 (Bézier surface)

Suppose that we are given a set of $(n+1)\cdot(m+1)$ control points in $\mathbb{R}^3$, with $0 \le i \le n$ and $0 \le j \le m$, where the control point on the i-th row and j-th column is denoted by $p_{i,j}$. The *Bézier surface* $\mathcal{S}\colon [0,1]\times[0,1] \to \mathbb{R}^3$ defined by $p_{i,j}$ is given by

$$\mathcal{S}(u,v) := \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v)p_{i,j} \qquad \text{for } (u,v)\in[0,1]\times[0,1],$$

where $B_{k,d}(x) := \binom{d}{k}x^k(1-x)^{d-k}$ is the k-th Bernstein basis polynomial of degree d.

Definition 114 (Bézier surface)

Suppose that we are given a set of $(n+1)\cdot(m+1)$ control points in $\mathbb{R}^3$, with $0 \le i \le n$ and $0 \le j \le m$, where the control point on the i-th row and j-th column is denoted by $p_{i,j}$. The *Bézier surface* $\mathcal{S}\colon [0,1]\times[0,1] \to \mathbb{R}^3$ defined by $p_{i,j}$ is given by

$$\mathcal{S}(u,v) := \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v)p_{i,j} \qquad \text{for } (u,v) \in [0,1]\times[0,1],$$

where $B_{k,d}(x) := \binom{d}{k}x^k(1-x)^{d-k}$ is the k-th Bernstein basis polynomial of degree d.

- Since $B_{i,n}(u)$ and $B_{j,m}(v)$ are polynomials of degree n and m, this is called a *Bézier surface of degree* (n,m).
- The set of control points is called a *Bézier net* or *control net*.

Lemma 115

For all $n, m \in \mathbb{N}_0$ and all $0 \leq i \leq n$ and $0 \leq j \leq m$, and all $(u, v) \in [0, 1] \times [0, 1]$, the term $B_{i,n}(u)B_{j,m}(v)$ is non-negative.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Surfaces

Lemma 115

For all $n, m \in \mathbb{N}_0$ and all $0 \leq i \leq n$ and $0 \leq j \leq m$, and all $(u, v) \in [0, 1] \times [0, 1]$, the term $B_{i,n}(u)B_{j,m}(v)$ is non-negative.

Lemma 116 (Partition of unity)

For all $m, n \in \mathbb{N}_0$, the sum of all $B_{i,n}(u)B_{j,m}(v)$ is one:

$$\sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = 1 \qquad \text{for all } (u, v) \in [0, 1] \times [0, 1].$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Surfaces

Lemma 115

For all $n, m \in \mathbb{N}_0$ and all $0 \le i \le n$ and $0 \le j \le m$, and all $(u, v) \in [0, 1] \times [0, 1]$, the term $B_{i,n}(u)B_{j,m}(v)$ is non-negative.

Lemma 116 (Partition of unity)

For all $m, n \in \mathbb{N}_0$, the sum of all $B_{i,n}(u)B_{j,m}(v)$ is one:

$$\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = 1 \qquad \text{for all } (u, v) \in [0, 1] \times [0, 1].$$

Proof: We have for all $m, n \in \mathbb{N}_0$ and all $(u, v) \in [0, 1] \times [0, 1]$

$$\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = \sum_{i=0}^{n} B_{i,n}(u) \left(\sum_{j=0}^{m} B_{j,m}(v)\right)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Surfaces

Lemma 115

For all $n, m \in \mathbb{N}_0$ and all $0 \le i \le n$ and $0 \le j \le m$, and all $(u, v) \in [0, 1] \times [0, 1]$, the term $B_{i,n}(u)B_{j,m}(v)$ is non-negative.

Lemma 116 (Partition of unity)

For all $m, n \in \mathbb{N}_0$, the sum of all $B_{i,n}(u)B_{j,m}(v)$ is one:

$$\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = 1 \qquad \text{for all } (u, v) \in [0, 1] \times [0, 1].$$

Proof: We have for all $m, n \in \mathbb{N}_0$ and all $(u, v) \in [0, 1] \times [0, 1]$

$$\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = \sum_{i=0}^{n} B_{i,n}(u)\left(\sum_{j=0}^{m} B_{j,m}(v)\right) = \sum_{i=0}^{n} B_{i,n}(u) = 1.$$

□

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Surfaces

Lemma 117 (Convex hull property)

A Bézier surface lies completely inside the convex hull of its control points.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 117 (Convex hull property)

A Bézier surface lies completely inside the convex hull of its control points.

Proof : Recall that $\mathcal{S}(u, v)$ is the linear combination of all its control points with non-negative coefficients whose sum is one. □

Properties of Bézier Surfaces

Lemma 117 (Convex hull property)

A Bézier surface lies completely inside the convex hull of its control points.

Proof : Recall that $\mathcal{S}(u, v)$ is the linear combination of all its control points with non-negative coefficients whose sum is one. □

Lemma 118

A Bézier surface passes through the four corners $p_{0,0}, p_{n,0}, p_{0,m}$ and $p_{n,m}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Surfaces

Lemma 117 (Convex hull property)

A Bézier surface lies completely inside the convex hull of its control points.

Proof : Recall that $\mathcal{S}(u, v)$ is the linear combination of all its control points with non-negative coefficients whose sum is one. □

Lemma 118

A Bézier surface passes through the four corners $p_{0,0}, p_{n,0}, p_{0,m}$ and $p_{n,m}$.

Proof : Recall that

$$B_{i,n}(0) = \begin{cases} 1 & \text{for } i = 0, \\ 0 & \text{for } i > 0, \end{cases} \qquad \text{and} \qquad B_{j,m}(0) = \begin{cases} 1 & \text{for } j = 0, \\ 0 & \text{for } j > 0. \end{cases}$$

Hence, $\mathcal{S}(0, 0) = B_{0,n}(0)B_{0,m}(0)p_{0,0} = p_{0,0}$. Similarly for the other corners. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of Bézier Surfaces

Lemma 117 (Convex hull property)

A Bézier surface lies completely inside the convex hull of its control points.

Proof : Recall that $\mathcal{S}(u, v)$ is the linear combination of all its control points with non-negative coefficients whose sum is one. □

Lemma 118

A Bézier surface passes through the four corners $p_{0,0}, p_{n,0}, p_{0,m}$ and $p_{n,m}$.

Proof : Recall that

$$B_{i,n}(0) = \begin{cases} 1 & \text{for } i = 0, \\ 0 & \text{for } i > 0, \end{cases} \quad \text{and} \quad B_{j,m}(0) = \begin{cases} 1 & \text{for } j = 0, \\ 0 & \text{for } j > 0. \end{cases}$$

Hence, $\mathcal{S}(0,0) = B_{0,n}(0)B_{0,m}(0)p_{0,0} = p_{0,0}$. Similarly for the other corners. □

Lemma 119

Applying an affine transformation to the control points results in the same transformation as obtained by transforming the surface's equation.

Lemma 120

Consider a Bézier surface $\mathcal{S}\colon [0,1]\times[0,1]\to\mathbb{R}^3$ defined by $(n+1)\cdot(m+1)$ control points $p_{i,j}$, with $0\le i\le n$ and $0\le j\le m$, and let $v_0\in[0,1]$ be fixed. Then $\mathcal{C}\colon [0,1]\to\mathbb{R}^3$ defined as

$$\mathcal{C}(u) := \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v_0)p_{i,j} \qquad \text{for } u\in[0,1]$$

is a Bézier curve defined by the $n+1$ control points $q_0, q_1, \ldots, q_n \in \mathbb{R}^3$, where

$$q_i := \sum_{j=0}^{m} B_{j,m}(v_0)p_{i,j} \qquad \text{for } 0\le i\le n.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 120

Consider a Bézier surface $\mathcal{S}\colon [0,1] \times [0,1] \to \mathbb{R}^3$ defined by $(n+1)\cdot(m+1)$ control points $p_{i,j}$, with $0 \le i \le n$ and $0 \le j \le m$, and let $v_0 \in [0,1]$ be fixed. Then $\mathcal{C}\colon [0,1] \to \mathbb{R}^3$ defined as

$$\mathcal{C}(u) := \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v_0)p_{i,j} \qquad \text{for } u \in [0,1]$$

is a Bézier curve defined by the $n+1$ control points $q_0, q_1, \ldots, q_n \in \mathbb{R}^3$, where

$$q_i := \sum_{j=0}^{m} B_{j,m}(v_0)p_{i,j} \qquad \text{for } 0 \le i \le n.$$

Proof: We have for all $u \in [0,1]$

$$\mathcal{C}(u) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v_0)p_{i,j} = \sum_{i=0}^{n} B_{i,n}(u)\left(\sum_{j=0}^{m} B_{j,m}(v_0)p_{i,j}\right) = \sum_{i=0}^{n} B_{i,n}(u)q_i.$$

Isoparametric Curves of Bézier Surfaces

Lemma 120

Consider a Bézier surface $\mathcal{S}\colon [0,1] \times [0,1] \to \mathbb{R}^3$ defined by $(n+1)\cdot(m+1)$ control points $p_{i,j}$, with $0 \le i \le n$ and $0 \le j \le m$, and let $v_0 \in [0,1]$ be fixed. Then $\mathcal{C}\colon [0,1] \to \mathbb{R}^3$ defined as

$$\mathcal{C}(u) := \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u) B_{j,m}(v_0) p_{i,j} \qquad \text{for } u \in [0,1]$$

is a Bézier curve defined by the $n+1$ control points $q_0, q_1, \ldots, q_n \in \mathbb{R}^3$, where

$$q_i := \sum_{j=0}^{m} B_{j,m}(v_0) p_{i,j} \qquad \text{for } 0 \le i \le n.$$

Proof: We have for all $u \in [0,1]$

$$\mathcal{C}(u) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u) B_{j,m}(v_0) p_{i,j} = \sum_{i=0}^{n} B_{i,n}(u) \left(\sum_{j=0}^{m} B_{j,m}(v_0) p_{i,j} \right) = \sum_{i=0}^{n} B_{i,n}(u) q_i.$$

- Analogously for fixed u_0.

Corollary 121

The *boundary curves* of a Bézier surface are Bézier curves defined by the boundary points of its control net.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Corollary 121

The *boundary curves* of a Bézier surface are Bézier curves defined by the boundary points of its control net.

Lemma 122 (Tangency in the corner points)

Consider a Bézier surface $\mathcal{S}\colon [0,1] \times [0,1] \to \mathbb{R}^3$ defined by $(n+1) \cdot (m+1)$ control points $p_{i,j}$, with $0 \leq i \leq n$ and $0 \leq j \leq m$. The tangent plane at $\mathcal{S}(0,0) = p_{0,0}$ is spanned by the vectors $p_{1,0} - p_{0,0}$ and $p_{0,1} - p_{0,0}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bézier Surface as Tensor-Product Surface

- A Bézier surface is generated by “multiplying” two Bézier curves: *tensor product surface*.

Lemma 123

Consider a Bézier surface $\mathcal{S}\colon [0,1]\times[0,1]\to\mathbb{R}^3$ defined by $(n+1)\cdot(m+1)$ control points $p_{i,j}$, with $0\le i\le n$ and $0\le j\le m$. Then $\mathcal{S}$ is a tensor-product surface:

$$\mathcal{S}(u,v) = (B_{0,n}(u), B_{1,n}(u), \ldots, B_{n,n}(u))\cdot \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,m} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,0} & p_{n,1} & \cdots & p_{n,m} \end{pmatrix} \cdot \begin{pmatrix} B_{0,m}(v) \\ B_{1,m}(v) \\ \vdots \\ B_{m,m}(v) \end{pmatrix}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bézier Surface as Tensor-Product Surface

- A Bézier surface is generated by "multiplying" two Bézier curves: *tensor product surface*.

Lemma 123

Consider a Bézier surface $\mathcal{S}\colon [0,1] \times [0,1] \to \mathbb{R}^3$ defined by $(n+1)\cdot(m+1)$ control points $p_{i,j}$, with $0 \le i \le n$ and $0 \le j \le m$. Then $\mathcal{S}$ is a tensor-product surface:

$$\mathcal{S}(u,v) = (B_{0,n}(u), B_{1,n}(u), \dots, B_{n,n}(u)) \cdot \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,m} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,0} & p_{n,1} & \cdots & p_{n,m} \end{pmatrix} \cdot \begin{pmatrix} B_{0,m}(v) \\ B_{1,m}(v) \\ \vdots \\ B_{m,m}(v) \end{pmatrix}$$

Proof : Just do the math! □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bézier Surface as Tensor-Product Surface

- A Bézier surface is generated by “multiplying” two Bézier curves: *tensor product surface*.

Lemma 123

Consider a Bézier surface $\mathcal{S}\colon [0,1]\times[0,1]\to\mathbb{R}^3$ defined by $(n+1)\cdot(m+1)$ control points $p_{i,j}$, with $0\le i\le n$ and $0\le j\le m$. Then $\mathcal{S}$ is a tensor-product surface:

$$\mathcal{S}(u,v) = (B_{0,n}(u), B_{1,n}(u), \ldots, B_{n,n}(u))\cdot \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,m} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,0} & p_{n,1} & \cdots & p_{n,m} \end{pmatrix} \cdot \begin{pmatrix} B_{0,m}(v) \\ B_{1,m}(v) \\ \vdots \\ B_{m,m}(v) \end{pmatrix}$$

Proof : Just do the math! □

- This can be re-written in matrix representation for $B_{i,n}(u)$ and $B_{j,m}(v)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Famous Bézier Surface Model: Utah Teapot

- The Utah teapot was designed in 1974 by Martin Newell at the Univ. of Utah.

[Image credits: https://en.wikipedia.org/wiki/Utah_teapot]

- The Utah teapot was designed in 1974 by Martin Newell at the Univ. of Utah.
- It is a hand-crafted Bézier model of a “Haushaltsteekanne” (“household teapot”) sold by Friesland Porzellan, at that time part of the German Melitta group.

[Image credits: https://en.wikipedia.org/wiki/Utah_teapot]

Famous Bézier Surface Model: Utah Teapot

- The Utah teapot was designed in 1974 by Martin Newell at the Univ. of Utah.
- It is a hand-crafted Bézier model of a "Haushaltsteekanne" ("household teapot") sold by Friesland Porzellan, at that time part of the German Melitta group.
- It is defined by 306 vertices and 32 Bézier patches.

[Image credits: https://en.wikipedia.org/wiki/Utah_teapot]

- The Utah teapot was designed in 1974 by Martin Newell at the Univ. of Utah.
- It is a hand-crafted Bézier model of a “Haushaltsteekanne” (“household teapot”) sold by Friesland Porzellan, at that time part of the German Melitta group.
- It has become one of the most iconic models. See, e.g., the "The Six Platonic Solids" by Arvo&Kirk (1987), showcasing “the newly discovered Teapotahedron”.
- It is defined by 306 vertices and 32 Bézier patches.

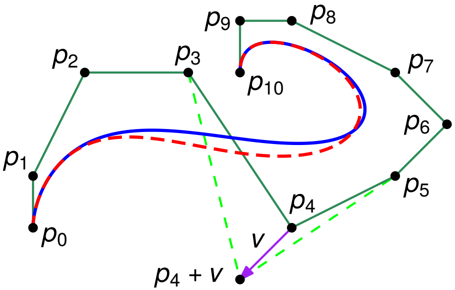
[Image credits: https://en.wikipedia.org/wiki/Utah_teapot]

4 B-Spline Curves and Surfaces

4 B-Spline Curves and Surfaces

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Modifying the vertex p_j of a Bézier curve causes a global change of the entire curve:

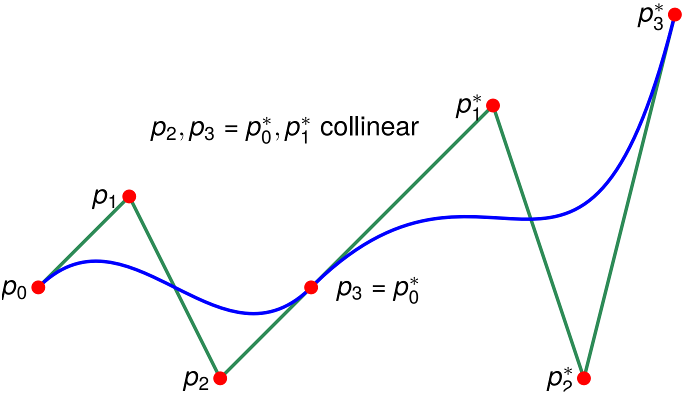


$$\mathcal{B}^{\star}(t) = \mathcal{B}(t) + B_{j,n}(t)v$$

- But $B_{j,n}(t) \neq 0$ for all t with $0 < t < 1$!

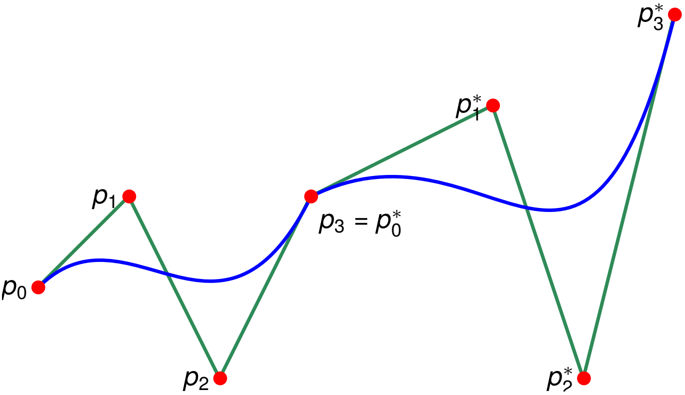
Shortcomings of Bézier Curves

- While it is easy to join two Bézier curves with G^1 continuity, achieving C^2 or even higher continuity is quite cumbersome.



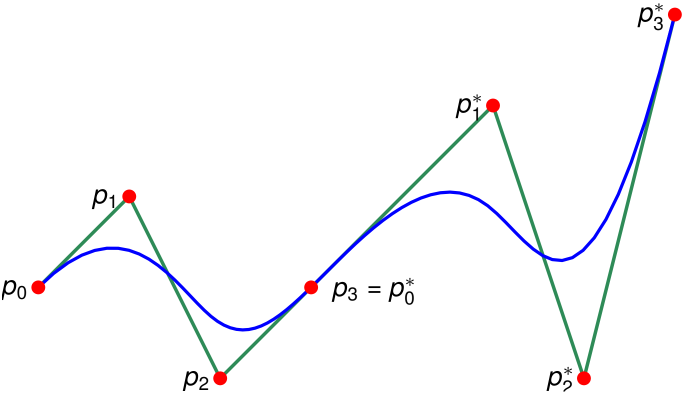
Shortcomings of Bézier Curves

- While it is easy to join two Bézier curves with G^1 continuity, achieving C^2 or even higher continuity is quite cumbersome.
- Even worse, changing the common end point of two consecutive Bézier curves destroys G^1 continuity.



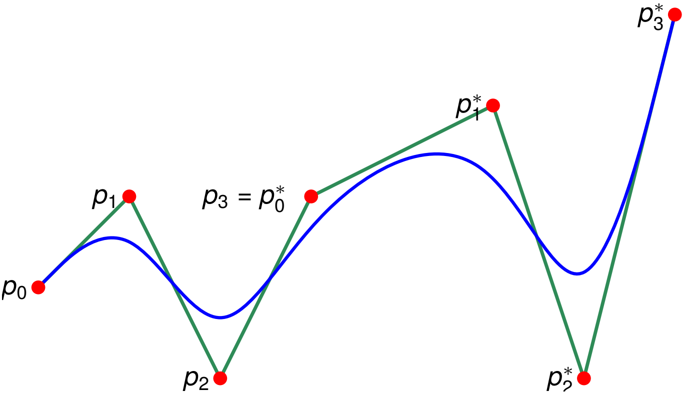
Shortcomings of Bézier Curves

- While it is easy to join two Bézier curves with G^1 continuity, achieving C^2 or even higher continuity is quite cumbersome.
- This will be easier for B-spline curves. (Depicted are two cubic B-splines.)



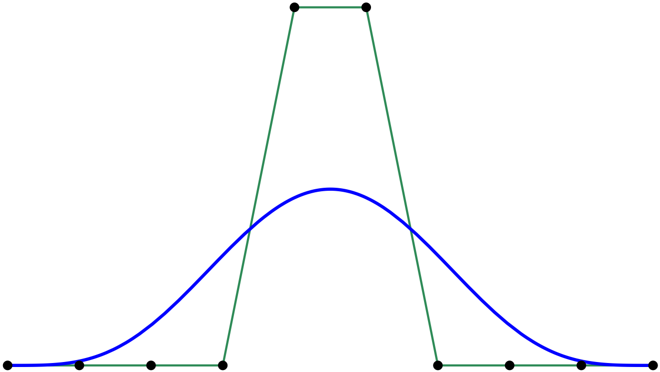
Shortcomings of Bézier Curves

- While it is easy to join two Bézier curves with G^1 continuity, achieving C^2 or even higher continuity is quite cumbersome.
- This will be easier for B-spline curves. (Depicted are two cubic B-splines.)



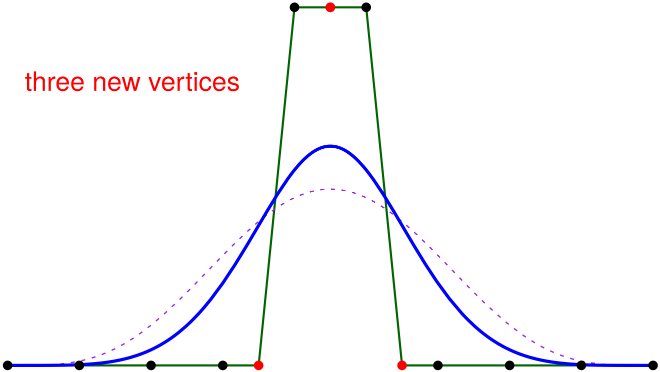
Shortcomings of Bézier Curves

- It is fairly difficult to squeeze a Bézier curve close to a sharp corner of the control polygon.



Shortcomings of Bézier Curves

- It is fairly difficult to squeeze a Bézier curve close to a sharp corner of the control polygon.
- Adding additional control vertices hardly helps but increases the degree of the Bézier curve, which may result in oscillation and cause numerical instability.



4 B-Spline Curves and Surfaces

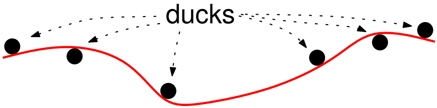
- Shortcomings of Bézier Curves
- B-Spline Basis Functions
  - Definition
  - Sample Basis Functions
  - Properties
- B-Spline Curves
- B-Spline Surfaces
- Non-Uniform Rational B-Spline Curves and Surfaces

Introduction to B-Splines

- Curves consisting of just one segment have several drawbacks:
  - The number of control points is directly related to the degree.
  - Often a high polynomial degree is required to satisfy all constraints given.
  - Interactive shape design is inaccurate or requires high computational costs.

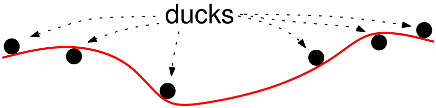
- Curves consisting of just one segment have several drawbacks:
  - The number of control points is directly related to the degree.
  - Often a high polynomial degree is required to satisfy all constraints given.
  - Interactive shape design is inaccurate or requires high computational costs.
- The solution is to use a sequence of polynomial or rational curves to form one continuous curve: *spline*.

- Curves consisting of just one segment have several drawbacks:
  - The number of control points is directly related to the degree.
  - Often a high polynomial degree is required to satisfy all constraints given.
  - Interactive shape design is inaccurate or requires high computational costs.
- The solution is to use a sequence of polynomial or rational curves to form one continuous curve: *spline*.
- Historically, the term *spline* (Dt.: Straklatte) was used for elastic wooden strips in the shipbuilding industry, which pass through given constrained points called *ducks* (Dt.: Molche) such that the strain of the strip is minimized.



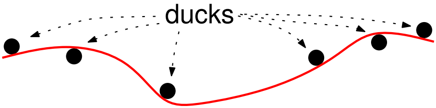
Introduction to B-Splines

- Curves consisting of just one segment have several drawbacks:
  - The number of control points is directly related to the degree.
  - Often a high polynomial degree is required to satisfy all constraints given.
  - Interactive shape design is inaccurate or requires high computational costs.
- The solution is to use a sequence of polynomial or rational curves to form one continuous curve: *spline*.
- Historically, the term *spline* (Dt.: Straklatte) was used for elastic wooden strips in the shipbuilding industry, which pass through given constrained points called *ducks* (Dt.: Molche) such that the strain of the strip is minimized.



- Mathematical splines were introduced by Isaac Jacob Schoenberg in 1946.

- Curves consisting of just one segment have several drawbacks:
  - The number of control points is directly related to the degree.
  - Often a high polynomial degree is required to satisfy all constraints given.
  - Interactive shape design is inaccurate or requires high computational costs.
- The solution is to use a sequence of polynomial or rational curves to form one continuous curve: *spline*.
- Historically, the term *spline* (Dt.: Straklatte) was used for elastic wooden strips in the shipbuilding industry, which pass through given constrained points called *ducks* (Dt.: Molche) such that the strain of the strip is minimized.



- Mathematical splines were introduced by Isaac Jacob Schoenberg in 1946.

Warning

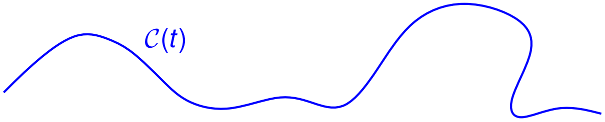
The terminology and the definitions used for B-splines vary from author to author! Thus, make sure to check *carefully* the definitions given in textbooks and papers.

Definition 124 (Spline)

A curve $\mathcal{C}\colon [a,b] \to \mathbb{R}^2$ is called a *spline* of degree k (and order $k+1$), for $k \in \mathbb{N}$, if there exist

- m polynomials $P_1, P_2, \ldots, P_m$ of degree k, for some $m \in \mathbb{N}$, and
- $m+1$ parameters $t_0, \ldots, t_m \in \mathbb{R}$

such that



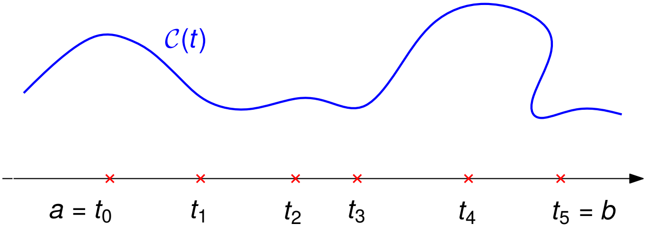
Definition 124 (Spline)

A curve $\mathcal{C}\colon [a, b] \to \mathbb{R}^2$ is called a *spline* of degree k (and order $k+1$), for $k \in \mathbb{N}$, if there exist

- m polynomials $P_1, P_2, \ldots, P_m$ of degree k, for some $m \in \mathbb{N}$, and
- $m+1$ parameters $t_0, ..., t_m \in \mathbb{R}$

such that

1. $a = t_0 \leq t_1 \leq \ldots \leq t_{m-1} \leq t_m = b$,



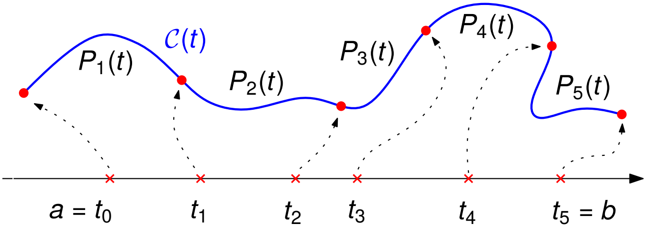
Definition 124 (Spline)

A curve $\mathcal{C} : [a, b] \to \mathbb{R}^2$ is called a *spline* of degree k (and order $k+1$), for $k \in \mathbb{N}$, if there exist

- m polynomials $P_1, P_2, \ldots, P_m$ of degree k, for some $m \in \mathbb{N}$, and
- $m+1$ parameters $t_0, \ldots, t_m \in \mathbb{R}$

such that

1. $a = t_0 \leq t_1 \leq \ldots \leq t_{m-1} \leq t_m = b$,
2. $\mathcal{C}|_{[t_{i-1},t_i]} = P_i|_{[t_{i-1},t_i]}$ for all $i \in \{1, 2, \ldots, m\}$.



- The numbers $t_0, ..., t_m$ are called *breakpoints* or *knots*.
- In general we expect $t_i < t_{i+1}$.
- The definition implies

$$P_i(t_i) = P_{i+1}(t_i) \quad \text{for all } i \in \{1, 2, \ldots, m-1\}.$$

- The numbers $t_0, ..., t_m$ are called *breakpoints* or *knots*.
- In general we expect $t_i < t_{i+1}$.
- The definition implies

$$P_i(t_i) = P_{i+1}(t_i) \quad \text{for all } i \in \{1, 2, \ldots, m-1\}.$$

- Special case $k = 1$: We get a polygonal curve.

- The numbers $t_0, ..., t_m$ are called *breakpoints* or *knots*.
- In general we expect $t_i < t_{i+1}$.
- The definition implies

$$P_i(t_i) = P_{i+1}(t_i) \quad \text{for all } i \in \{1, 2, \ldots, m-1\}.$$

- Special case $k = 1$: We get a polygonal curve.

- The polynomials join with some unknown degree of continuity at the breakpoints. (We have at least C^0-continuity.)
- Obvious problem: How can we achieve a reasonable degree of continuity?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Vector

Definition 125 (Knot vector, Dt.: Knotenvektor)

In general, a knot vector is a sequence of non-decreasing real numbers ("*knots*").

Knot Vector

Definition 125 (Knot vector, Dt.: Knotenvektor)

In general, a knot vector is a sequence of non-decreasing real numbers ("*knots*"). A *finite knot vector* is a sequence of $m+1$ real numbers $\tau := (t_0, t_1, t_2, \ldots, t_m)$, for some $m \in \mathbb{N}$, such that $t_i \leq t_{i+1}$ for all $0 \leq i < m$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 125 (Knot vector, Dt.: Knotenvektor)

In general, a knot vector is a sequence of non-decreasing real numbers ("*knots*").
A *finite knot vector* is a sequence of $m+1$ real numbers $\tau := (t_0, t_1, t_2, \ldots, t_m)$, for some $m \in \mathbb{N}$, such that $t_i \leq t_{i+1}$ for all $0 \leq i < m$.
An *infinite knot vector* is an infinite sequence of real numbers $\tau := (t_0, t_1, t_2, \ldots)$ such that $t_i \leq t_{i+1}$ for all $i \in \mathbb{N}_0$.

Knot Vector

Definition 125 (Knot vector, Dt.: Knotenvektor)

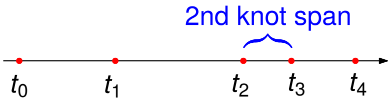
In general, a knot vector is a sequence of non-decreasing real numbers ("*knots*").
A *finite knot vector* is a sequence of $m+1$ real numbers $\tau := (t_0, t_1, t_2, \ldots, t_m)$, for some $m \in \mathbb{N}$, such that $t_i \leq t_{i+1}$ for all $0 \leq i < m$.
An *infinite knot vector* is an infinite sequence of real numbers $\tau := (t_0, t_1, t_2, \ldots)$ such that $t_i \leq t_{i+1}$ for all $i \in \mathbb{N}_0$.
A *bi-infinite knot vector* is a bi-infinite sequence of real numbers $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ such that $t_i \leq t_{i+1}$ for all $i \in \mathbb{Z}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Vector

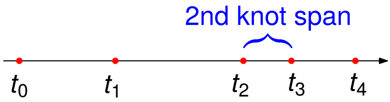
Definition 125 (Knot vector, Dt.: Knotenvektor)

In general, a knot vector is a sequence of non-decreasing real numbers ("*knots*").
A *finite knot vector* is a sequence of $m+1$ real numbers $\tau := (t_0, t_1, t_2, \ldots, t_m)$, for some $m \in \mathbb{N}$, such that $t_i \leq t_{i+1}$ for all $0 \leq i < m$.
An *infinite knot vector* is an infinite sequence of real numbers $\tau := (t_0, t_1, t_2, \ldots)$ such that $t_i \leq t_{i+1}$ for all $i \in \mathbb{N}_0$.
A *bi-infinite knot vector* is a bi-infinite sequence of real numbers $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ such that $t_i \leq t_{i+1}$ for all $i \in \mathbb{Z}$.
The *i-th knot span* is given by the (half-open) interval $[t_i, t_{i+1}[\subset \mathbb{R}$.



Definition 125 (Knot vector, Dt.: Knotenvektor)

In general, a knot vector is a sequence of non-decreasing real numbers ("*knots*").
A *finite knot vector* is a sequence of $m+1$ real numbers $\tau := (t_0, t_1, t_2, \ldots, t_m)$, for some $m \in \mathbb{N}$, such that $t_i \le t_{i+1}$ for all $0 \le i < m$.
An *infinite knot vector* is an infinite sequence of real numbers $\tau := (t_0, t_1, t_2, \ldots)$ such that $t_i \le t_{i+1}$ for all $i \in \mathbb{N}_0$.
A *bi-infinite knot vector* is a bi-infinite sequence of real numbers $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ such that $t_i \le t_{i+1}$ for all $i \in \mathbb{Z}$.
The *i-th knot span* is given by the (half-open) interval $[t_i, t_{i+1}[\subset \mathbb{R}$.



- For (bi)infinite knot vectors we assume $\sup_{i \to \infty} t_i = \infty$ and $\inf_{i \to -\infty} t_i = -\infty$.
- For some of the subsequent definitions we will find it convenient to deal with (bi)infinite knot vectors. With some extra care for "boundary conditions" one could replace all (bi)infinite knot vectors by finite knot vectors.

Knot Vector

Definition 126 (Multiplicity of a knot, Dt.: Vielfachheit eines Knotens)

Let τ be a finite or (bi)infinite knot vector. If a knot t_i appears exactly $k > 1$ times in τ, for a permissible value of $i \in \mathbb{Z}$, i.e., if $t_{i-1} < t_i = t_{i+1} = \cdots = t_{i+k-1} < t_{i+k}$, then t_i is a *multiple knot* of *multiplicity* k. Otherwise, if t_i appears only once in τ then t_i is a *simple knot*.

Knot Vector

Definition 126 (Multiplicity of a knot, Dt.: Vielfachheit eines Knotens)

Let τ be a finite or (bi)infinite knot vector. If a knot t_i appears exactly $k > 1$ times in τ, for a permissible value of $i \in \mathbb{Z}$, i.e., if $t_{i-1} < t_i = t_{i+1} = \cdots = t_{i+k-1} < t_{i+k}$, then t_i is a *multiple knot* of *multiplicity* k. Otherwise, if t_i appears only once in τ then t_i is a *simple knot*.

Definition 127 (Uniform knot vector)

A finite or (bi)infinite knot vector is *uniform* if there exists $c \in \mathbb{R}^+$ such that $t_{i+1} - t_i = c$ for all (permissible) values of $i \in \mathbb{Z}$, except for possibly the first and last knots of higher multiplicity in case of a finite knot vector. Otherwise, the knot vector is *non-uniform*.

B-Spline Basis Functions

- We define the B-spline basis functions analytically, using the recurrence relation found independently by de Boor and Mansfield (1972) and Cox (1972).

Definition 128 (B-spline basis function)

Let τ be a finite or (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the *i*-th *B-spline basis function*, $N_{i,k,\tau}(t)$, of *degree k* (and *order* $k+1$) relative to τ is defined as,

- We define the B-spline basis functions analytically, using the recurrence relation found independently by de Boor and Mansfield (1972) and Cox (1972).

Definition 128 (B-spline basis function)

Let τ be a finite or (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the *i*-th *B-spline basis function*, $N_{i,k,\tau}(t)$, of *degree* k (and *order* $k+1$) relative to τ is defined as,
if $k = 0$,

$$N_{i,0,\tau}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- We define the B-spline basis functions analytically, using the recurrence relation found independently by de Boor and Mansfield (1972) and Cox (1972).

Definition 128 (B-spline basis function)

Let τ be a finite or (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the i-th *B-spline basis function*, $N_{i,k,\tau}(t)$, of *degree* k (and *order* $k+1$) relative to τ is defined as,
if $k = 0$,

$$N_{i,0,\tau}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

and if $k > 0$ as

$$N_{i,k,\tau}(t) = \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1,\tau}(t).$$

B-Spline Basis Functions

- We define the B-spline basis functions analytically, using the recurrence relation found independently by de Boor and Mansfield (1972) and Cox (1972).

Definition 128 (B-spline basis function)

Let τ be a finite or (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the i-th *B-spline basis function*, $N_{i,k,\tau}(t)$, of *degree* k (and *order* $k+1$) relative to τ is defined as,
if $k = 0$,

$$N_{i,0,\tau}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

and if $k > 0$ as

$$N_{i,k,\tau}(t) = \frac{t - t_i}{t_{i+k} - t_i}\, N_{i,k-1,\tau}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}}\, N_{i+1,k-1,\tau}(t).$$

- In case of multiple knots, indeterminate terms of the form $\frac{0}{0}$ are taken as zero!
- Alternatively, one can demand $t_i < t_{i+k}$ for all (permissible) $i \in \mathbb{Z}$.
- Aka: *Normalized B(asic)-Spline Blending Functions*.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Plugging into the definition yields

$$N_{i,1,\tau}(t) = \frac{t - t_i}{t_{i+1} - t_i} N_{i,0,\tau}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} N_{i+1,0,\tau}(t)$$

$$= \begin{cases} 0 & \text{if } t \notin [t_i, t_{i+2}[, \\ \frac{t-t_i}{t_{i+1}-t_i} & \text{if } t \in [t_i, t_{i+1}[, \\ \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} & \text{if } t \in [t_{i+1}, t_{i+2}[. \end{cases}$$

B-Spline Basis Functions

- Plugging into the definition yields

$$N_{i,1,\tau}(t) = \frac{t - t_i}{t_{i+1} - t_i} N_{i,0,\tau}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} N_{i+1,0,\tau}(t)$$

$$= \begin{cases} 0 & \text{if } t \notin [t_i, t_{i+2}[, \\ \frac{t-t_i}{t_{i+1}-t_i} & \text{if } t \in [t_i, t_{i+1}[, \\ \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} & \text{if } t \in [t_{i+1}, t_{i+2}[. \end{cases}$$

- The functions $N_{i,1,\tau}(t)$ are called *hat functions* or *chapeau functions*. They are widely used in signal processing and finite-element techniques.
- Note that $N_{i,1,\tau}(t)$ is continuous at t_{i+1}.

- Plugging into the definition yields

$$N_{i,1,\tau}(t) = \frac{t - t_i}{t_{i+1} - t_i} N_{i,0,\tau}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} N_{i+1,0,\tau}(t)$$

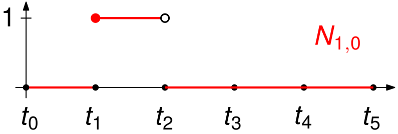
$$= \begin{cases} 0 & \text{if } t \notin [t_i, t_{i+2}[, \\ \frac{t-t_i}{t_{i+1}-t_i} & \text{if } t \in [t_i, t_{i+1}[, \\ \frac{t_{i+2}-t}{t_{i+2}-t_{i+1}} & \text{if } t \in [t_{i+1}, t_{i+2}[. \end{cases}$$

- The functions $N_{i,1,\tau}(t)$ are called *hat functions* or *chapeau functions*. They are widely used in signal processing and finite-element techniques.
- Note that $N_{i,1,\tau}(t)$ is continuous at t_{i+1}.
- For a uniform knot vector τ with $c := t_{i+1} - t_i$ this simplifies to

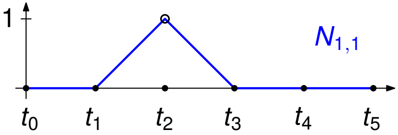
$$N_{i,1,\tau}(t) = \begin{cases} 0 & \text{if } t \notin [t_i, t_{i+2}[, \\ \frac{1}{c}(t - t_i) & \text{if } t \in [t_i, t_{i+1}[, \\ \frac{1}{c}(t_{i+2} - t) & \text{if } t \in [t_{i+1}, t_{i+2}[. \end{cases}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

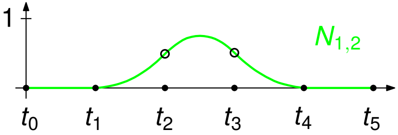
- Basis functions $N_{i,k,\tau}$.



$N_{i,0}$ is a step function that is 1 over the knot span $[t_i, t_{i+1}[$



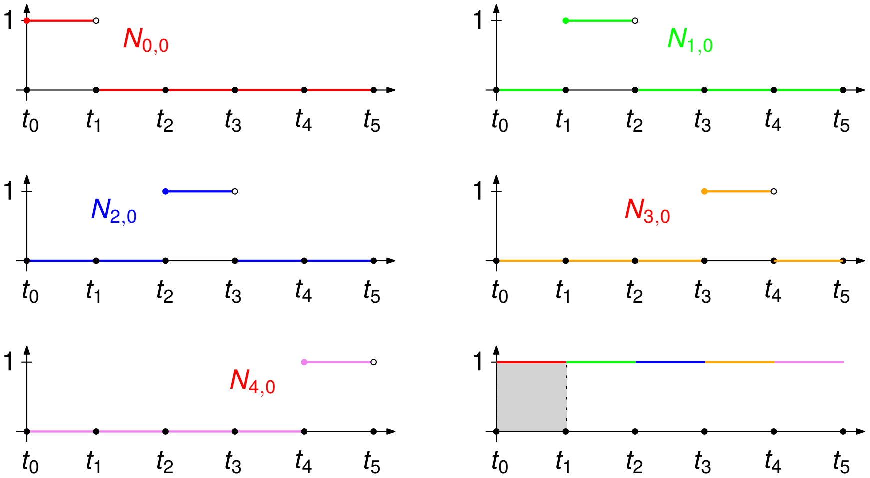
$N_{i,1}$ is a piecewise linear function that is non-zero over two knot spans $[t_i, t_{i+2}[$ and goes from 0 to 1 and back



$N_{i,2}$ is a piecewise quadratic function that is non-zero over three knot spans $[t_i, t_{i+3}[$

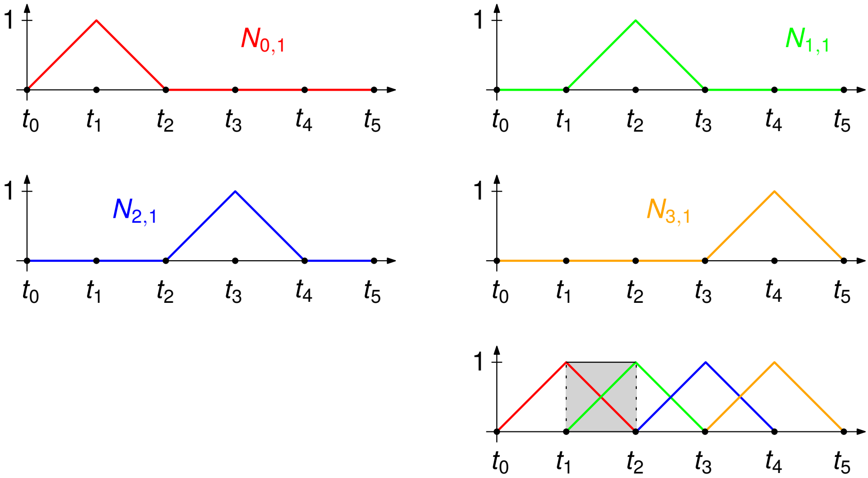
Sample B-Spline Basis Functions

- Basis functions of degree 0:



Sample B-Spline Basis Functions

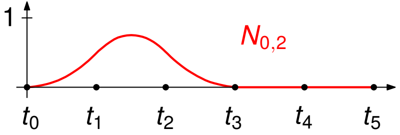
- Basis functions of degree 1:

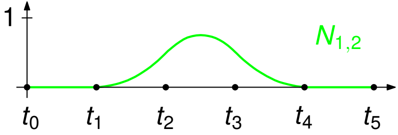


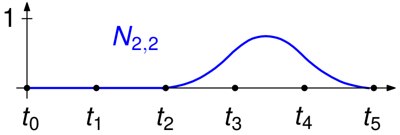
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

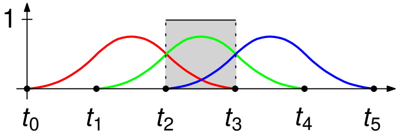
Sample B-Spline Basis Functions

- Basis functions of degree 2:



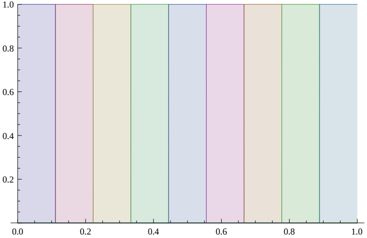




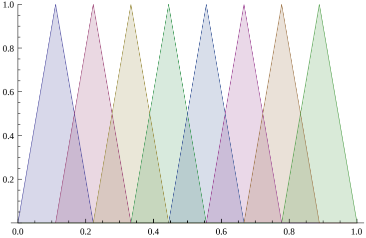


Sample B-Spline Basis Functions

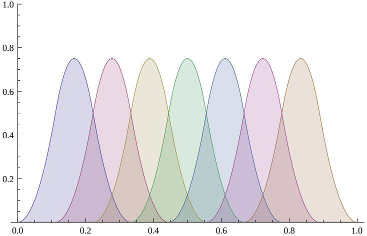
- Uniform knot vector $(0, \frac{1}{9}, \frac{2}{9}, \frac{3}{9}, \frac{4}{9}, \frac{5}{9}, \frac{6}{9}, \frac{7}{9}, \frac{8}{9}, 1)$ with ten knots.



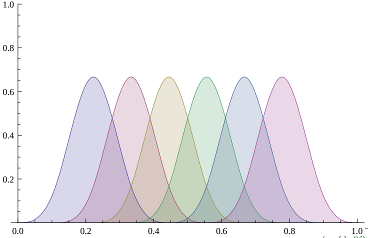
$N_{0,0}, N_{1,0}, \ldots, N_{8,0}$



$N_{0,1}, N_{1,1}, \ldots, N_{7,1}$



$N_{0,2}, N_{1,2}, \ldots, N_{6,2}$

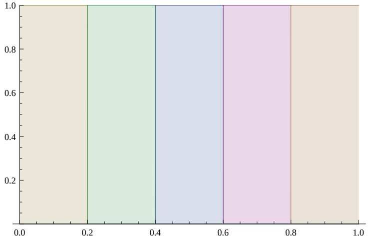


$N_{0,3}, N_{1,3}, \ldots, N_{5,3}$

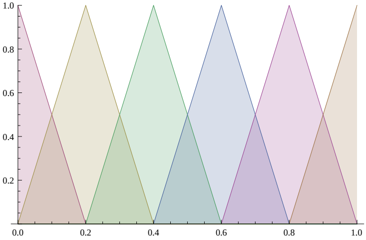
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Sample B-Spline Basis Functions

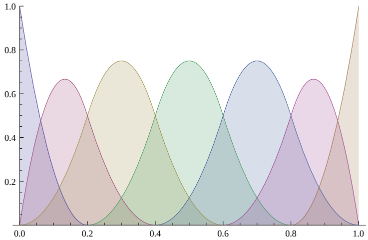
- Clamped uniform knot vector $(0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1)$ with ten knots.



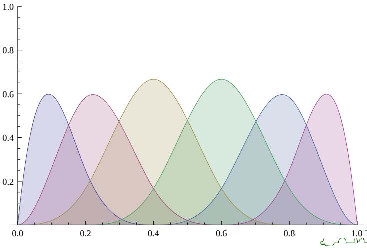
$N_{2,0}, N_{3,0}, \ldots, N_{6,0}$



$N_{1,1}, N_{2,1}, \ldots, N_{5,1}$



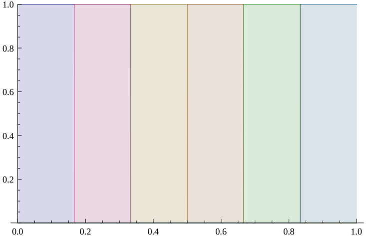
$N_{0,2}, N_{1,2}, \ldots, N_{6,2}$



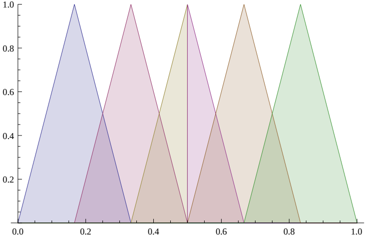
$N_{0,3}, N_{1,3}, \ldots, N_{5,3}$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

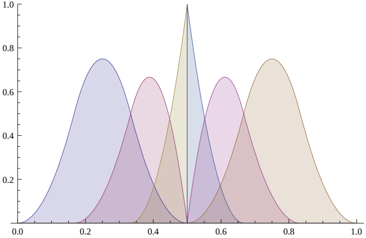
- Non-uniform knot vector $(0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1)$ with ten knots.



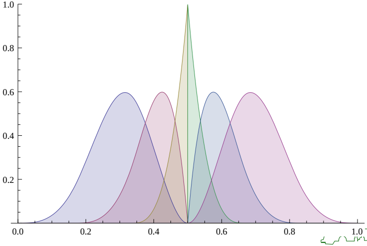
$N_{0,0}, N_{1,0}, N_{2,0}, N_{6,0}, N_{7,0}, N_{8,0}$



$N_{0,1}, N_{1,1}, N_{2,1}, N_{5,1}, N_{6,1}, N_{7,1}$



$N_{0,2}, N_{1,2}, N_{2,2}, N_{4,2}, N_{5,2}, N_{6,2}$



$N_{0,3}, N_{1,3}, \ldots, N_{5,3}$

Properties of B-Spline Basis Functions

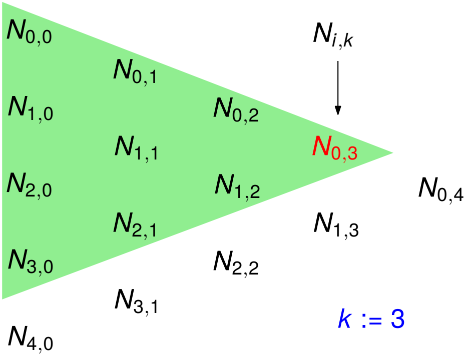
- It is common to omit the explicit mentioning of the dependency of $N_{i,k,\tau}(t)$ on τ, and to write $N_{i,k}(t)$. (And sometimes we simply write $N_{i,k}$ ...)

Properties of B-Spline Basis Functions

- It is common to omit the explicit mentioning of the dependency of $N_{i,k,\tau}(t)$ on τ, and to write $N_{i,k}(t)$. (And sometimes we simply write $N_{i,k}$ ...)
- For $k > 0$, each $N_{i,k,\tau}(t)$ is a linear combination of two B-spline basis functions of degree $k-1$: $N_{i,k-1,\tau}(t)$ and $N_{i+1,k-1,\tau}(t)$.
- This suggests a recursive analysis of the dependencies.

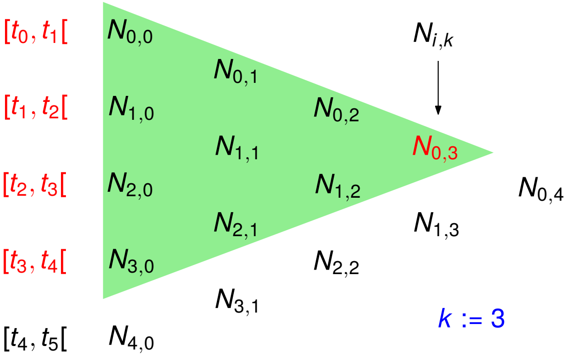
$$
\begin{array}{ccccc}
N_{0,0} & & & & \\
 & N_{0,1} & & & \\
N_{1,0} & & N_{0,2} & & \\
 & N_{1,1} & & N_{0,3} & \\
N_{2,0} & & N_{1,2} & & N_{0,4} \\
 & N_{2,1} & & N_{1,3} & \\
N_{3,0} & & N_{2,2} & & \\
 & N_{3,1} & & & \\
N_{4,0} & & & &
\end{array}
$$

- It is common to omit the explicit mentioning of the dependency of $N_{i,k,\tau}(t)$ on τ, and to write $N_{i,k}(t)$. (And sometimes we simply write $N_{i,k}$ ...)
- For $k > 0$, each $N_{i,k,\tau}(t)$ is a linear combination of two B-spline basis functions of degree $k-1$: $N_{i,k-1,\tau}(t)$ and $N_{i+1,k-1,\tau}(t)$.
- $N_{i,k,\tau}(t)$ depends on $N_{i,0,\tau}(t), N_{i+1,0,\tau}(t), \ldots, N_{i+k,0,\tau}(t)$.



Properties of B-Spline Basis Functions

- It is common to omit the explicit mentioning of the dependency of $N_{i,k,\tau}(t)$ on τ, and to write $N_{i,k}(t)$. (And sometimes we simply write $N_{i,k}$ ...)
- For $k > 0$, each $N_{i,k,\tau}(t)$ is a linear combination of two B-spline basis functions of degree $k-1$: $N_{i,k-1,\tau}(t)$ and $N_{i+1,k-1,\tau}(t)$.
- $N_{i,k,\tau}(t)$ is non-zero only for $t \in [t_i, t_{i+k+1}[$.



Lemma 129 (Local support, Dt.: lokaler Träger)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have

$$N_{i,k,\tau}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Properties of B-Spline Basis Functions

Lemma 129 (Local support, Dt.: lokaler Träger)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have

$$N_{i,k,\tau}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Proof : We do a proof by induction on k.

I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.

Lemma 129 (Local support, Dt.: lokaler Träger)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have

$$N_{i,k,\tau}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Proof : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.
I.H.: Suppose that it is true for all basis functions of degree $k - 1$, for some arbitrary but fixed $k \in \mathbb{N}$. I.e., $N_{i,k-1,\tau}(t) = 0$ if $t \notin [t_i, t_{i+k}[$, for all (permissible) $i \in \mathbb{Z}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 129 (Local support, Dt.: lokaler Träger)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have

$$N_{i,k,\tau}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Proof: We do a proof by induction on k.

I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.

I.H.: Suppose that it is true for all basis functions of degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$. I.e., $N_{i,k-1,\tau}(t) = 0$ if $t \notin [t_i, t_{i+k}[$, for all (permissible) $i \in \mathbb{Z}$.

I.S.: Recall that

$$N_{i,k,\tau}(t) = \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1,\tau}(t).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of B-Spline Basis Functions

Lemma 129 (Local support, Dt.: lokaler Träger)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have

$$N_{i,k,\tau}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Proof: We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.
I.H.: Suppose that it is true for all basis functions of degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$. I.e., $N_{i,k-1,\tau}(t) = 0$ if $t \notin [t_i, t_{i+k}[$, for all (permissible) $i \in \mathbb{Z}$.
I.S.: Recall that

$$N_{i,k,\tau}(t) = \frac{t - t_i}{t_{i+k} - t_i}\, N_{i,k-1,\tau}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}}\, N_{i+1,k-1,\tau}(t).$$

Hence, $N_{i,k,\tau}(t) = 0$ if $t \notin ([t_i, t_{i+k}[\cup [t_{i+1}, t_{i+k+1}[)$, i.e., if $t \notin [t_i, t_{i+k+1}[$. □

Lemma 130 (Non-negativity)

We have $N_{i,k,\tau}(t) \geq 0$ for all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, and all real t.

Lemma 130 (Non-negativity)

We have $N_{i,k,\tau}(t) \geq 0$ for all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, and all real t.

Proof : Again we do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of B-Spline Basis Functions

Lemma 130 (Non-negativity)

We have $N_{i,k,\tau}(t) \geq 0$ for all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, and all real t.

Proof : Again we do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.
I.H.: Suppose that it is true for all basis functions of degree $k - 1$, for some arbitrary but fixed $k \in \mathbb{N}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 130 (Non-negativity)

We have $N_{i,k,\tau}(t) \geq 0$ for all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, and all real t.

Proof: Again we do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.
I.H.: Suppose that it is true for all basis functions of degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$.
I.S.: Lemma 129 tells us that $N_{i,k,\tau}(t) = 0$ if $t \notin [t_i, t_{i+k+1}[$. Hence, we can focus on $t \in [t_i, t_{i+k+1}[$ and get

$$N_{i,k,\tau}(t) = \underbrace{\frac{t - t_i}{t_{i+k} - t_i}}_{\geq 0 \text{ for } t \in [t_i, t_{i+k+1}[} \cdot \underbrace{N_{i,k-1,\tau}(t)}_{\geq 0 \text{ (I.H.)}} + \underbrace{\frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}}}_{\geq 0 \text{ for } t \in [t_i, t_{i+k+1}[} \cdot \underbrace{N_{i+1,k-1,\tau}(t)}_{\geq 0 \text{ (I.H.)}}$$
$$\geq 0.$$

□

Lemma 130 (Non-negativity)

We have $N_{i,k,\tau}(t) \geq 0$ for all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, and all real t.

Proof : Again we do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$ and all (permissible) $i \in \mathbb{Z}$.
I.H.: Suppose that it is true for all basis functions of degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$.
I.S.: Lemma 129 tells us that $N_{i,k,\tau}(t) = 0$ if $t \notin [t_i, t_{i+k+1}[$. Hence, we can focus on $t \in [t_i, t_{i+k+1}[$ and get

$$N_{i,k,\tau}(t) = \underbrace{\frac{t - t_i}{t_{i+k} - t_i}}_{\geq 0 \text{ for } t \in [t_i, t_{i+k+1}[} \cdot \underbrace{N_{i,k-1,\tau}(t)}_{\geq 0 \text{ (I.H.)}} + \underbrace{\frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}}}_{\geq 0 \text{ for } t \in [t_i, t_{i+k+1}[} \cdot \underbrace{N_{i+1,k-1,\tau}(t)}_{\geq 0 \text{ (I.H.)}}$$
$$\geq 0.$$

□

Lemma 131

For all $k \in \mathbb{N}$, all B-spline basis functions of degree k are continuous.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of B-Spline Basis Functions

Lemma 132 (Local influence)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the basis functions

$$N_{i-k,k,\tau}(t),\ N_{i-k+1,k,\tau}(t),\ \ldots\ ,\ N_{i,k,\tau}(t)$$

are the only (at most) $k+1$ basis functions of degree k that are (possibly) non-zero over the interval $[t_i, t_{i+1}[$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

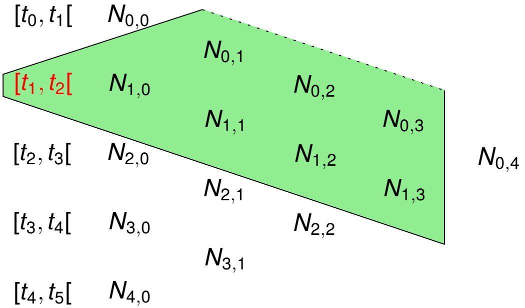
Properties of B-Spline Basis Functions

Lemma 132 (Local influence)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the basis functions

$$N_{i-k,k,\tau}(t),\ N_{i-k+1,k,\tau}(t),\ \ldots\ ,\ N_{i,k,\tau}(t)$$

are the only (at most) $k+1$ basis functions of degree k that are (possibly) non-zero over the interval $[t_i, t_{i+1}[$.



Lemma 132 (Local influence)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the basis functions

$$N_{i-k,k,\tau}(t),\; N_{i-k+1,k,\tau}(t),\; \ldots\;,\; N_{i,k,\tau}(t)$$

are the only (at most) $k+1$ basis functions of degree k that are (possibly) non-zero over the interval $[t_i, t_{i+1}[$.

Proof : The Local Support Lemma 129 tells us that

$$N_{j,k,\tau}(t) = 0 \qquad \text{if } t \notin [t_j, t_{j+k+1}[$$

and, thus, possibly non-zero only if $t \in [t_j, t_{j+k+1}[$.

Properties of B-Spline Basis Functions

Lemma 132 (Local influence)

Let $\tau := (\ldots, t_{-2}, t_{-1}, t_0, t_1, t_2, \ldots)$ be a (bi)infinite knot vector. For all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$, the basis functions

$$N_{i-k,k,\tau}(t),\ N_{i-k+1,k,\tau}(t),\ \ldots\ ,\ N_{i,k,\tau}(t)$$

are the only (at most) $k+1$ basis functions of degree k that are (possibly) non-zero over the interval $[t_i, t_{i+1}[$.

Proof: The Local Support Lemma 129 tells us that

$$N_{j,k,\tau}(t) = 0 \qquad \text{if } t \notin [t_j, t_{j+k+1}[$$

and, thus, possibly non-zero only if $t \in [t_j, t_{j+k+1}[$.
Hence, $N_{j,k,\tau}(t) \neq 0$ over $[t_i, t_{i+1}[$ only if $i \geq j$ and $i+1 \leq j+k+1$, i.e., if $j \leq i$ and $j \geq i-k$. Thus,

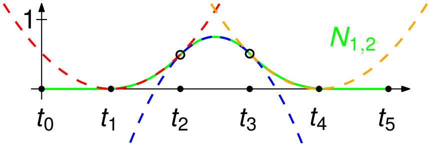
$$N_{i-k,k,\tau}(t),\ N_{i-k+1,k,\tau}(t),\ \ldots\ ,\ N_{i,k,\tau}(t)$$

are the only B-spline basis functions that are (possibly) non-zero over $[t_i, t_{i+1}[$. □

Properties of B-Spline Basis Functions

Lemma 133

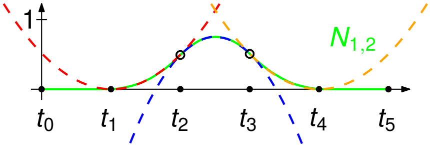
For all $k \in \mathbb{N}_0$, all B-spline basis functions of degree k are piecewise polynomials of degree k.



Properties of B-Spline Basis Functions

Lemma 133

For all $k \in \mathbb{N}_0$, all B-spline basis functions of degree k are piecewise polynomials of degree k.



Lemma 134

For all $k \in \mathbb{N}$, all B-spline basis functions of degree k are $k - r$ times continuously differentiable at a knot of multiplicity r, and $k - 1$ times continuously differentiable everywhere else. The first derivative of $N_{i,k}(t)$ is given as follows:

$$N'_{i,k}(t) = \frac{k}{t_{i+k} - t_i} N_{i,k-1}(t) - \frac{k}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 135

For a uniform knot vector τ, all B-spline basis functions of the same degree are shifted copies of each other: For all $t \in \mathbb{R}$ and all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have $N_{i,k,\tau}(t) = N_{0,k,\tau}(t - i \cdot c)$, where $c := t_1 - t_0$.

Properties of B-Spline Basis Functions

Lemma 135

For a uniform knot vector τ, all B-spline basis functions of the same degree are shifted copies of each other: For all $t \in \mathbb{R}$ and all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have $N_{i,k,\tau}(t) = N_{0,k,\tau}(t - i \cdot c)$, where $c := t_1 - t_0$.

Lemma 136 (Partition of unity, Dt.: Zerlegung der Eins)

Let $\tau = (t_0, t_1, t_2, \ldots, t_m)$ be a finite knot vector, and $k \in \mathbb{N}_0$ with $k < \frac{m}{2}$. Then,

$$\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) = 1 \qquad \text{for all } t \in [t_k, t_{m-k}[.$$

Properties of B-Spline Basis Functions

Lemma 135

For a uniform knot vector τ, all B-spline basis functions of the same degree are shifted copies of each other: For all $t \in \mathbb{R}$ and all (permissible) $i \in \mathbb{Z}$ and $k \in \mathbb{N}_0$ we have $N_{i,k,\tau}(t) = N_{0,k,\tau}(t - i \cdot c)$, where $c := t_1 - t_0$.

Lemma 136 (Partition of unity, Dt.: Zerlegung der Eins)

Let $\tau = (t_0, t_1, t_2, \ldots, t_m)$ be a finite knot vector, and $k \in \mathbb{N}_0$ with $k < \frac{m}{2}$. Then,

$$\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) = 1 \qquad \text{for all } t \in [t_k, t_{m-k}[.$$

Corollary 137

Let $\tau = (t_0, t_1, t_2, \ldots, t_{n+k+1})$ be a finite knot vector, for some $k \in \mathbb{N}_0$ with $k \leq n$. Then,

$$\sum_{i=0}^{n} N_{i,k,\tau}(t) = 1 \qquad \text{for all } t \in [t_k, t_{n+1}[.$$

Properties of B-Spline Basis Functions

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.

Properties of B-Spline Basis Functions

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.
I.S.: Recall that (by Lem. 129)

$$N_{0,k-1,\tau}(t) = 0 \quad \text{for} \quad t \notin [t_0, t_k[\qquad \text{and} \qquad N_{m-k,k-1,\tau}(t) = 0 \quad \text{for} \quad t \notin [t_{m-k}, t_m[.$$

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.
I.S.: Recall that (by Lem. 129)

$$N_{0,k-1,\tau}(t) = 0 \ \text{ for } \ t \notin [t_0, t_k[\qquad \text{and} \qquad N_{m-k,k-1,\tau}(t) = 0 \ \text{ for } \ t \notin [t_{m-k}, t_m[.$$

Let $t \in [t_k, t_{m-k}[$ be arbitrary but fixed. Applying the recursion yields

$$\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) = \sum_{i=0}^{m-k-1} \left(\frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \right)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Proof of Lemma 136 (Partition of Unity): We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.
I.S.: Recall that (by Lem. 129)

$$N_{0,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_0, t_k[\qquad \text{and} \qquad N_{m-k,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_{m-k}, t_m[.$$

Let $t \in [t_k, t_{m-k}[$ be arbitrary but fixed. Applying the recursion yields

$$\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) = \sum_{i=0}^{m-k-1} \left(\frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1,\tau}(t) \right)$$

$$= \sum_{i=1}^{m-k-1} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1,\tau}(t) + \sum_{i=0}^{m-k-2} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1,\tau}(t)$$

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.
I.S.: Recall that (by Lem. 129)

$$N_{0,k-1,\tau}(t) = 0 \text{ for } t \notin [t_0, t_k[\qquad \text{and} \qquad N_{m-k,k-1,\tau}(t) = 0 \text{ for } t \notin [t_{m-k}, t_m[.$$

Let $t \in [t_k, t_{m-k}[$ be arbitrary but fixed. Applying the recursion yields

$$\begin{aligned}
\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) &= \sum_{i=0}^{m-k-1} \left(\frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \right) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=0}^{m-k-2} \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=1}^{m-k-1} \frac{t_{i+k}-t}{t_{i+k}-t_i} N_{i,k-1,\tau}(t)
\end{aligned}$$

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.
I.S.: Recall that (by Lem. 129)

$$N_{0,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_0, t_k[\qquad \text{and} \qquad N_{m-k,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_{m-k}, t_m[.$$

Let $t \in [t_k, t_{m-k}[$ be arbitrary but fixed. Applying the recursion yields

$$\begin{aligned}
\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) &= \sum_{i=0}^{m-k-1} \left(\frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \right) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=0}^{m-k-2} \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=1}^{m-k-1} \frac{t_{i+k}-t}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) \\
&= \sum_{i=1}^{m-k-1} N_{i,k-1,\tau}(t)
\end{aligned}$$

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.
I.S.: Recall that (by Lem. 129)

$$N_{0,k-1,\tau}(t) = 0 \text{ for } t \notin [t_0, t_k[\qquad \text{and} \qquad N_{m-k,k-1,\tau}(t) = 0 \text{ for } t \notin [t_{m-k}, t_m[.$$

Let $t \in [t_k, t_{m-k}[$ be arbitrary but fixed. Applying the recursion yields

$$\begin{aligned}
\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) &= \sum_{i=0}^{m-k-1} \left(\frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \right) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=0}^{m-k-2} \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=1}^{m-k-1} \frac{t_{i+k}-t}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) \\
&= \sum_{i=1}^{m-k-1} N_{i,k-1,\tau}(t) = \sum_{i=0}^{m-k} N_{i,k-1,\tau}(t)
\end{aligned}$$

Proof of Lemma 136 (Partition of Unity) : We do a proof by induction on k.
I.B.: By definition, this claim is correct for $k := 0$.
I.H.: Suppose that it is true for degree $k-1$, for some arbitrary but fixed $k \in \mathbb{N}$ such that $k < \frac{m}{2}$. I.e., suppose that $\sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) = 1$ for all $t \in [t_{k-1}, t_{m-k+1}[$.
I.S.: Recall that (by Lem. 129)

$$N_{0,k-1,\tau}(t) = 0 \text{ for } t \notin [t_0, t_k[\qquad \text{and} \qquad N_{m-k,k-1,\tau}(t) = 0 \text{ for } t \notin [t_{m-k}, t_m[.$$

Let $t \in [t_k, t_{m-k}[$ be arbitrary but fixed. Applying the recursion yields

$$\begin{aligned}
\sum_{i=0}^{m-k-1} N_{i,k,\tau}(t) &= \sum_{i=0}^{m-k-1} \left(\frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \right) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=0}^{m-k-2} \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1,\tau}(t) \\
&= \sum_{i=1}^{m-k-1} \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) + \sum_{i=1}^{m-k-1} \frac{t_{i+k}-t}{t_{i+k}-t_i} N_{i,k-1,\tau}(t) \\
&= \sum_{i=1}^{m-k-1} N_{i,k-1,\tau}(t) = \sum_{i=0}^{m-k} N_{i,k-1,\tau}(t) \overset{t \in [t_{k-1}, t_{m-k+1}[}{=} 1.
\end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 138 (B-spline curve)

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then the *B-spline curve* of *degree* k (and *order* $k+1$) relative to τ with control points $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{P}(t) := \sum_{i=0}^{n} N_{i,k,\tau}(t) p_i \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where $N_{i,k,\tau}$ is the i-th B-spline basis function of degree k relative to τ.

Definition 138 (B-spline curve)

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then the *B-spline curve* of *degree* k (and *order* $k+1$) relative to τ with control points $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{P}(t) := \sum_{i=0}^{n} N_{i,k,\tau}(t) p_i \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where $N_{i,k,\tau}$ is the i-th B-spline basis function of degree k relative to τ.

- The degree k is (except for $k \leq n$) independent of the number $n+1$ of control points!
- The restriction of t to the interval $[t_k, t_{n+1}[$ guarantees that the basis functions sum up to 1 for all (permissible) values of t. (Recall the Partition of Unity, Cor. 137.)

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Clamped and Unclamped B-Spline Curves

Definition 139 (Clamped B-spline)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$, over the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. If $t_0 = t_1 = \ldots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \ldots = t_{n+k+1}$ then we say that the knot vector and the B-spline curve are *clamped*.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 139 (Clamped B-spline)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$, over the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. If $t_0 = t_1 = \ldots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \ldots = t_{n+k+1}$ then we say that the knot vector and the B-spline curve are *clamped*.

- Recall that the Partition of Unity (Cor. 137) holds for all $t \in [t_k, t_{n+1}[$.
- Typically, for a clamped knot vector,

$$0 = t_0 = t_1 = \ldots = t_k \qquad \text{and} \qquad t_{n+1} = \ldots = t_{n+k+1} = 1.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 139 (Clamped B-spline)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$, over the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. If $t_0 = t_1 = \ldots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \ldots = t_{n+k+1}$ then we say that the knot vector and the B-spline curve are *clamped*.

- Recall that the Partition of Unity (Cor. 137) holds for all $t \in [t_k, t_{n+1}[$.
- Typically, for a clamped knot vector,

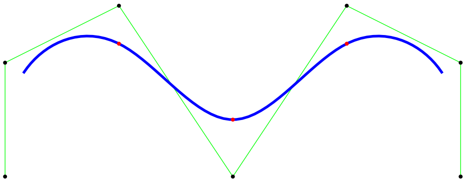
$$0 = t_0 = t_1 = \ldots = t_k \qquad \text{and} \qquad t_{n+1} = \ldots = t_{n+k+1} = 1.$$

Lemma 140

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$, over the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then $\mathcal{P}$ starts in p_0 and ends in p_n.

Clamped and Unclamped B-Spline Curves

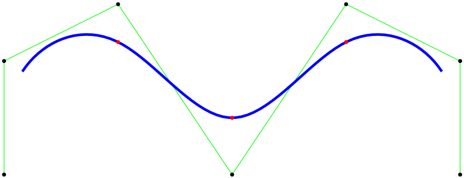
- Control points: $\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$.



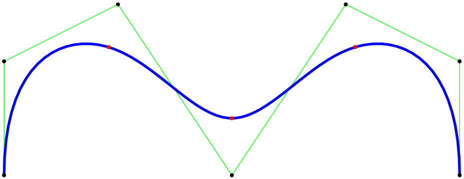
uniform unclamped cubic B-spline: $\tau = (0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1)$

Clamped and Unclamped B-Spline Curves

- Control points: $\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$.



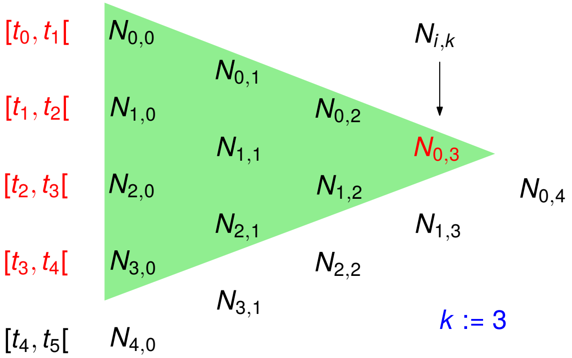
uniform unclamped cubic B-spline: $\tau = (0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1)$



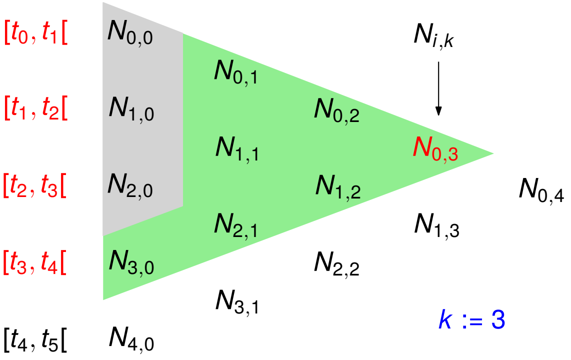
uniform clamped cubic B-spline: $\tau = (0, 0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1, 1)$.

Proof of Lemma 140

We prove $\mathcal{P}(t_k) = p_0$. Recall that $N_{0,k}(t)$ is non-zero only for $t \in [t_0, t_{k+1}[$.



We prove $\mathcal{P}(t_k) = p_0$. Recall that $N_{0,k}(t)$ is non-zero only for $t \in [t_0, t_{k+1}[$.



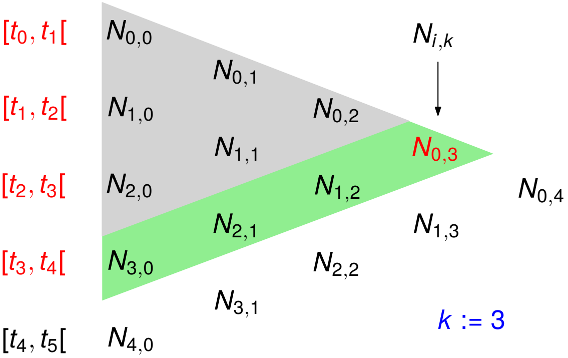
However, for a clamped knot vector with $t_0 = t_1 = \ldots = t_k < t_{k+1}$ we have

$$N_{0,0}(t) = N_{1,0}(t) = \ldots = N_{k-1,0}(t) = 0 \quad \text{for all } t, \quad \text{and } N_{k,0}(t_k) = 1.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Proof of Lemma 140

We prove $\mathcal{P}(t_k) = p_0$. Recall that $N_{0,k}(t)$ is non-zero only for $t \in [t_0, t_{k+1}[$.



However, for a clamped knot vector with $t_0 = t_1 = \ldots = t_k < t_{k+1}$ we have

$$N_{0,0}(t) = N_{1,0}(t) = \ldots = N_{k-1,0}(t) = 0 \quad \text{for all } t, \quad \text{and } N_{k,0}(t_k) = 1.$$

The recursion formula for the B-spline basis functions yields

$$N_{i,j}(t) = 0 \quad \text{for all } i, j \text{ with } i + j \leq k - 1 \text{ and for all } t.$$

Proof of Lemma 140

Applying the standard recursion for the B-spline basis functions at parameter t_k,

$$N_{i,k}(t_k) = \frac{t_k - t_i}{t_{i+k} - t_i}\, N_{i,k-1}(t_k) + \frac{t_{i+k+1} - t_k}{t_{i+k+1} - t_{i+1}}\, N_{i+1,k-1}(t_k),$$

for $i := 0$ (and subsequently for $i := j$ and $k - j$, for $j \in \{1, \ldots, k-1\}$) yields

$$N_{0,k}(t_k) = \frac{t_k - t_0}{t_k - t_0}\, N_{0,k-1}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_1}\, N_{1,k-1}(t_k)$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Proof of Lemma 140

Applying the standard recursion for the B-spline basis functions at parameter t_k,

$$N_{i,k}(t_k) = \frac{t_k - t_i}{t_{i+k} - t_i}\, N_{i,k-1}(t_k) + \frac{t_{i+k+1} - t_k}{t_{i+k+1} - t_{i+1}}\, N_{i+1,k-1}(t_k),$$

for $i := 0$ (and subsequently for $i := j$ and $k - j$, for $j \in \{1, \ldots, k-1\}$) yields

$$\begin{aligned} N_{0,k}(t_k) &= \frac{t_k - t_0}{t_k - t_0}\, N_{0,k-1}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_1}\, N_{1,k-1}(t_k) \\ &= \frac{t_{k+1} - t_k}{t_{k+1} - t_k}\, N_{1,k-1}(t_k) = N_{1,k-1}(t_k) \end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Proof of Lemma 140

Applying the standard recursion for the B-spline basis functions at parameter t_k,

$$N_{i,k}(t_k) = \frac{t_k - t_i}{t_{i+k} - t_i}\, N_{i,k-1}(t_k) + \frac{t_{i+k+1} - t_k}{t_{i+k+1} - t_{i+1}}\, N_{i+1,k-1}(t_k),$$

for $i := 0$ (and subsequently for $i := j$ and $k - j$, for $j \in \{1, \ldots, k-1\}$) yields

$$\begin{aligned}
N_{0,k}(t_k) &= \frac{t_k - t_0}{t_k - t_0}\, N_{0,k-1}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_1}\, N_{1,k-1}(t_k) \\
&= \frac{t_{k+1} - t_k}{t_{k+1} - t_k}\, N_{1,k-1}(t_k) = N_{1,k-1}(t_k) \\
&= \frac{t_k - t_1}{t_k - t_1}\, N_{1,k-2}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_2}\, N_{2,k-2}(t_k)
\end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Proof of Lemma 140

Applying the standard recursion for the B-spline basis functions at parameter t_k,

$$N_{i,k}(t_k) = \frac{t_k - t_i}{t_{i+k} - t_i}\, N_{i,k-1}(t_k) + \frac{t_{i+k+1} - t_k}{t_{i+k+1} - t_{i+1}}\, N_{i+1,k-1}(t_k),$$

for $i := 0$ (and subsequently for $i := j$ and $k - j$, for $j \in \{1, \ldots, k-1\}$) yields

$$\begin{aligned}
N_{0,k}(t_k) &= \frac{t_k - t_0}{t_k - t_0}\, N_{0,k-1}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_1}\, N_{1,k-1}(t_k) \\
&= \frac{t_{k+1} - t_k}{t_{k+1} - t_k}\, N_{1,k-1}(t_k) = N_{1,k-1}(t_k) \\
&= \frac{t_k - t_1}{t_k - t_1}\, N_{1,k-2}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_2}\, N_{2,k-2}(t_k) \\
&= N_{2,k-2}(t_k)
\end{aligned}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Proof of Lemma 140

Applying the standard recursion for the B-spline basis functions at parameter t_k,

$$N_{i,k}(t_k) = \frac{t_k - t_i}{t_{i+k} - t_i}\, N_{i,k-1}(t_k) + \frac{t_{i+k+1} - t_k}{t_{i+k+1} - t_{i+1}}\, N_{i+1,k-1}(t_k),$$

for $i := 0$ (and subsequently for $i := j$ and $k - j$, for $j \in \{1, \ldots, k-1\}$) yields

$$\begin{aligned} N_{0,k}(t_k) &= \frac{t_k - t_0}{t_k - t_0}\, N_{0,k-1}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_1}\, N_{1,k-1}(t_k) \\ &= \frac{t_{k+1} - t_k}{t_{k+1} - t_k}\, N_{1,k-1}(t_k) = N_{1,k-1}(t_k) \\ &= \frac{t_k - t_1}{t_k - t_1}\, N_{1,k-2}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_2}\, N_{2,k-2}(t_k) \\ &= N_{2,k-2}(t_k) = \cdots = N_{k,0}(t_k) \\ &= 1. \end{aligned}$$

Proof of Lemma 140

Applying the standard recursion for the B-spline basis functions at parameter t_k,

$$N_{i,k}(t_k) = \frac{t_k - t_i}{t_{i+k} - t_i} N_{i,k-1}(t_k) + \frac{t_{i+k+1} - t_k}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t_k),$$

for $i := 0$ (and subsequently for $i := j$ and $k - j$, for $j \in \{1, \ldots, k-1\}$) yields

$$\begin{aligned}
N_{0,k}(t_k) &= \frac{t_k - t_0}{t_k - t_0} N_{0,k-1}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_1} N_{1,k-1}(t_k) \\
&= \frac{t_{k+1} - t_k}{t_{k+1} - t_k} N_{1,k-1}(t_k) = N_{1,k-1}(t_k) \\
&= \frac{t_k - t_1}{t_k - t_1} N_{1,k-2}(t_k) + \frac{t_{k+1} - t_k}{t_{k+1} - t_2} N_{2,k-2}(t_k) \\
&= N_{2,k-2}(t_k) = \cdots = N_{k,0}(t_k) \\
&= 1.
\end{aligned}$$

Hence, due to the Partition of Unity, Cor. 137, $N_{i,k}(t_k) = 0$ for $i > 0$ and we get

$$\sum_{i=0}^{n} N_{i,k}(t_k) p_i = N_{0,k}(t_k) p_0 = p_0.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Generation of Knot Vector

- Suppose that a B-spline curve over $[0, 1]$ has $n + 1$ control points $p_0, p_1, \ldots, p_n$ and degree k.

Generation of Knot Vector

- Suppose that a B-spline curve over $[0, 1]$ has $n + 1$ control points $p_0, p_1, \ldots, p_n$ and degree k.
- We need $m + 1$ knots, where $m = n + k + 1$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Generation of Knot Vector

- Suppose that a B-spline curve over $[0, 1]$ has $n + 1$ control points $p_0, p_1, \ldots, p_n$ and degree k.
- We need $m + 1$ knots, where $m = n + k + 1$.
- If the B-spline curve is clamped then we get

$$t_0 = t_1 = \ldots = t_k = 0 \quad \text{and} \quad t_{n+1} = t_{n+2} = \ldots = t_{n+k+1} = 1.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Generation of Knot Vector

- Suppose that a B-spline curve over $[0, 1]$ has $n+1$ control points $p_0, p_1, \ldots, p_n$ and degree k.
- We need $m+1$ knots, where $m = n+k+1$.
- If the B-spline curve is clamped then we get

$$t_0 = t_1 = \ldots = t_k = 0 \quad \text{and} \quad t_{n+1} = t_{n+2} = \ldots = t_{n+k+1} = 1.$$

- The remaining $n-k$ knots can be spaced uniformly or non-uniformly.
- For uniformly spaced internal knots the interval $[0, 1]$ is divided into $n-k+1$ subintervals. In this case the knots are given as follows:

$$t_0 = t_1 = \ldots = t_k = 0$$

$$t_{k+j} = \frac{j}{n-k+1} \qquad \text{for } j = 1, 2, \ldots, n-k$$

$$t_{n+1} = t_{n+2} = \ldots = t_{n+k+1} = 1$$

Generation of Knot Vector

- Suppose that $n := 6$, i.e., that we have seven control points $p_0, \ldots, p_6$, and want to construct a clamped cubic B-spline curve. (Hence, $k = 3$.)

Generation of Knot Vector

- Suppose that $n := 6$, i.e., that we have seven control points $p_0, \ldots, p_6$, and want to construct a clamped cubic B-spline curve. (Hence, $k = 3$.)
- We have in total $m + 1 = n + k + 2 = 6 + 3 + 2 = 11$ knots and get

$$\tau := (\underbrace{0, 0, 0, 0}_{k+1=4}, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, \underbrace{1, 1, 1, 1}_{k+1=4})$$

as uniform knot vector.

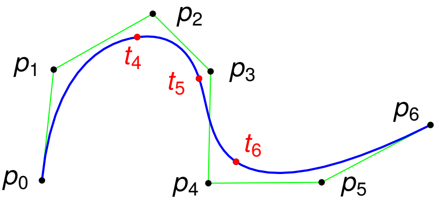
Generation of Knot Vector

- Suppose that $n := 6$, i.e., that we have seven control points $p_0, \ldots, p_6$, and want to construct a clamped cubic B-spline curve. (Hence, $k = 3$.)
- We have in total $m + 1 = n + k + 2 = 6 + 3 + 2 = 11$ knots and get

$$\tau := (\underbrace{0, 0, 0, 0}_{k+1=4}, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, \underbrace{1, 1, 1, 1}_{k+1=4})$$

as uniform knot vector.

- For $(p_0, \ldots, p_6) := \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 5 \\ 0 \end{pmatrix}, \begin{pmatrix} 7 \\ 1 \end{pmatrix} \right)$ we get the following clamped, cubic and C^2-continuous B-spline curve:



Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof: The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

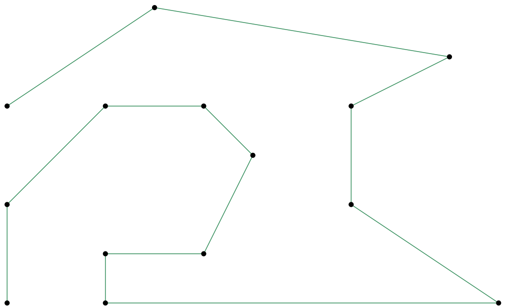
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof: The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

- Clamped uniform B-spline for a control polygon with 14 vertices:

$$\left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}1\\3\end{pmatrix},\begin{pmatrix}3\\5\end{pmatrix},\begin{pmatrix}5\\5\end{pmatrix},\begin{pmatrix}6\\4\end{pmatrix},\begin{pmatrix}5\\2\end{pmatrix},\begin{pmatrix}3\\2\end{pmatrix},\begin{pmatrix}3\\1\end{pmatrix},\begin{pmatrix}11\\1\end{pmatrix},\begin{pmatrix}8\\3\end{pmatrix},\begin{pmatrix}8\\5\end{pmatrix},\begin{pmatrix}10\\6\end{pmatrix},\begin{pmatrix}4\\7\end{pmatrix},\begin{pmatrix}1\\5\end{pmatrix}\right\}$$



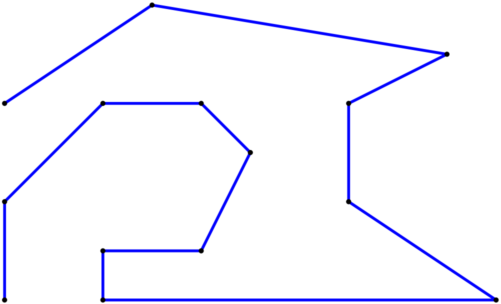
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof : The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

- Clamped uniform B-spline of degree 1 for a control polygon with 14 vertices:

$$\left\{ \binom{1}{1}, \binom{1}{3}, \binom{3}{5}, \binom{5}{5}, \binom{6}{4}, \binom{5}{2}, \binom{3}{2}, \binom{3}{1}, \binom{11}{1}, \binom{8}{3}, \binom{8}{5}, \binom{10}{6}, \binom{4}{7}, \binom{1}{5} \right\}$$



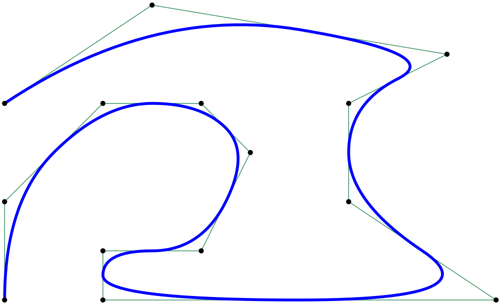
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof: The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

- Clamped uniform B-spline of degree 2 for a control polygon with 14 vertices:

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix} \right\}$$



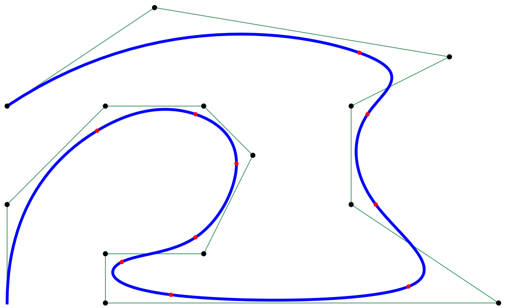
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof : The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

- Clamped uniform B-spline of degree 3 for a control polygon with 14 vertices:

$$\left\{\binom{1}{1},\binom{1}{3},\binom{3}{5},\binom{5}{5},\binom{6}{4},\binom{5}{2},\binom{3}{2},\binom{3}{1},\binom{11}{1},\binom{8}{3},\binom{8}{5},\binom{10}{6},\binom{4}{7},\binom{1}{5}\right\}$$



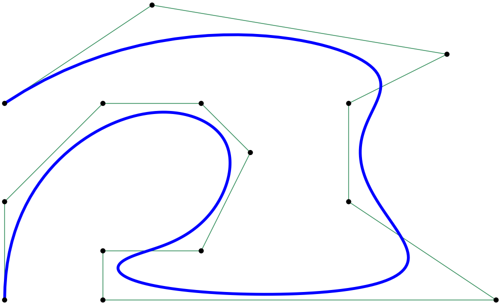
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof: The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

- Clamped uniform B-spline of degree 4 for a control polygon with 14 vertices:

$$\left\{ \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1\\3\end{pmatrix}, \begin{pmatrix}3\\5\end{pmatrix}, \begin{pmatrix}5\\5\end{pmatrix}, \begin{pmatrix}6\\4\end{pmatrix}, \begin{pmatrix}5\\2\end{pmatrix}, \begin{pmatrix}3\\2\end{pmatrix}, \begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}11\\1\end{pmatrix}, \begin{pmatrix}8\\3\end{pmatrix}, \begin{pmatrix}8\\5\end{pmatrix}, \begin{pmatrix}10\\6\end{pmatrix}, \begin{pmatrix}4\\7\end{pmatrix}, \begin{pmatrix}1\\5\end{pmatrix} \right\}$$



Properties of B-Spline Curves

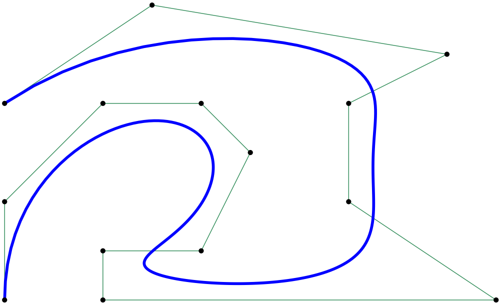
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof: The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

- Clamped uniform B-spline of degree 7 for a control polygon with 14 vertices:

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix} \right\}$$



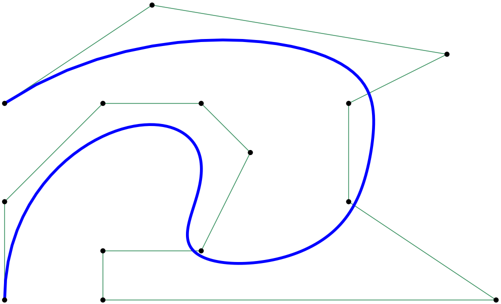
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof: The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. □

- Clamped uniform B-spline of degree 10 for a control polygon with 14 vertices:

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix} \right\}$$



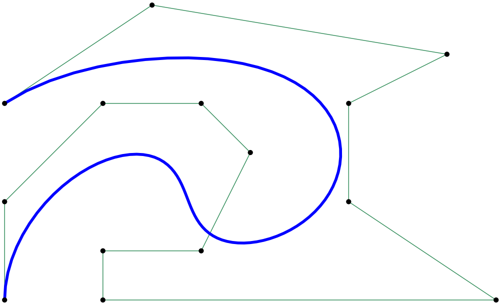
Lemma 141

The lower the degree of a B-spline curve, the closer it follows its control polygon.

Sketch of proof : The lower the degree, the fewer control points contribute to $\mathcal{P}(t)$. For $k := 1$ it is simply the convex combination of pairs of control points. $\square$

- Clamped uniform B-spline of degree 13 for a control polygon with 14 vertices:

$\left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}1\\3\end{pmatrix},\begin{pmatrix}3\\5\end{pmatrix},\begin{pmatrix}5\\5\end{pmatrix},\begin{pmatrix}6\\4\end{pmatrix},\begin{pmatrix}5\\2\end{pmatrix},\begin{pmatrix}3\\2\end{pmatrix},\begin{pmatrix}3\\1\end{pmatrix},\begin{pmatrix}11\\1\end{pmatrix},\begin{pmatrix}8\\3\end{pmatrix},\begin{pmatrix}8\\5\end{pmatrix},\begin{pmatrix}10\\6\end{pmatrix},\begin{pmatrix}4\\7\end{pmatrix},\begin{pmatrix}1\\5\end{pmatrix}\right\}$ — Bézier curve!



Lemma 142 (Variation diminishing property)

If a straight line intersects the control polygon of a B-spline curve m times then it intersects the actual B-spline curve at most m times.

Properties of B-Spline Curves

Lemma 142 (Variation diminishing property)

If a straight line intersects the control polygon of a B-spline curve m times then it intersects the actual B-spline curve at most m times.

Lemma 143 (Affine invariance)

Any B-spline representation is affinely invariant, i.e., given any affine map π, the image curve $\pi(\mathcal{P})$ of a B-spline curve $\mathcal{P}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of B-Spline Curves

Lemma 142 (Variation diminishing property)

If a straight line intersects the control polygon of a B-spline curve m times then it intersects the actual B-spline curve at most m times.

Lemma 143 (Affine invariance)

Any B-spline representation is affinely invariant, i.e., given any affine map π, the image curve $\pi(\mathcal{P})$ of a B-spline curve $\mathcal{P}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

Sketch of proof : The proof is identical to the proof of the affine invariance of Bézier curves, recall Lem. 106. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 144

Let $\mathcal{P}$ be a clamped B-spline curve of degree k over $[0, 1]$ defined by $k + 1$ control points with position vectors $p_0, p_1, \ldots, p_k$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{2k+1})$, for $k \in \mathbb{N}_0$. Then $\mathcal{P}$ is a Bézier curve of degree k.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of B-Spline Curves

Lemma 144

Let $\mathcal{P}$ be a clamped B-spline curve of degree k over $[0, 1]$ defined by $k + 1$ control points with position vectors $p_0, p_1, \ldots, p_k$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{2k+1})$, for $k \in \mathbb{N}_0$. Then $\mathcal{P}$ is a Bézier curve of degree k.

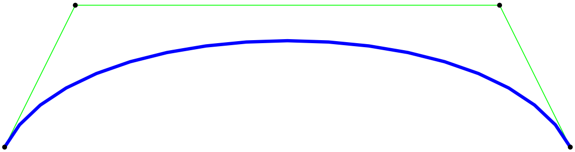
- Note: This implies $0 = t_0 = t_1 = \ldots = t_k$ and $1 = t_{k+1} = \ldots = t_{2k} = t_{2k+1}$.
- Of course, this lemma can also be formulated for a parameter interval other than $[0, 1]$.

Properties of B-Spline Curves

Lemma 144

Let $\mathcal{P}$ be a clamped B-spline curve of degree k over $[0, 1]$ defined by $k+1$ control points with position vectors $p_0, p_1, \ldots, p_k$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{2k+1})$, for $k \in \mathbb{N}_0$. Then $\mathcal{P}$ is a Bézier curve of degree k.

- Note: This implies $0 = t_0 = t_1 = \ldots = t_k$ and $1 = t_{k+1} = \ldots = t_{2k} = t_{2k+1}$.
- Of course, this lemma can also be formulated for a parameter interval other than $[0, 1]$.
- Clamped (uniform) B-spline of degree 3 for knot vector $(0, 0, 0, 0, 1, 1, 1, 1)$ and control polygon $\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix} \right\}$:



Lemma 145

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$, and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then

$$\mathcal{P}'(t) = \sum_{i=0}^{n-1} N_{i+1,k-1}(t) q_i \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where

$$q_i := \frac{k}{t_{i+k+1} - t_{i+1}} (p_{i+1} - p_i) \quad \text{for } i \in \{0, 1, \ldots, n-1\}$$

and the knot vector τ remains unchanged.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 145

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$, and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then

$$\mathcal{P}'(t) = \sum_{i=0}^{n-1} N_{i+1,k-1}(t) q_i \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where

$$q_i := \frac{k}{t_{i+k+1} - t_{i+1}}(p_{i+1} - p_i) \quad \text{for } i \in \{0, 1, \ldots, n-1\}$$

and the knot vector τ remains unchanged.

Sketch of proof: This is a consequence of Lem. 134 and some (lengthy) analysis. □

Lemma 146

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then, for the new knot vector $\tau' := (t_1, t_2, \ldots, t_{n+k-1}, t_{n+k})$,

$$\mathcal{P}'(t) = \sum_{i=0}^{n-1} N_{i,k-1,\tau'}(t) q_i \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where

$$q_i := \frac{k}{t_{i+k+1} - t_{i+1}} (p_{i+1} - p_i) \quad \text{for } i \in \{0, 1, \ldots, n-1\}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 146

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then, for the new knot vector $\tau' := (t_1, t_2, \ldots, t_{n+k-1}, t_{n+k})$,

$$\mathcal{P}'(t) = \sum_{i=0}^{n-1} N_{i,k-1,\tau'}(t) q_i \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where

$$q_i := \frac{k}{t_{i+k+1} - t_{i+1}} (p_{i+1} - p_i) \quad \text{for } i \in \{0, 1, \ldots, n-1\}.$$

Sketch of proof : One can show that $N_{i+1,k-1,\tau}(t)$ is equal to $N_{i,k-1,\tau'}(t)$ for all $t \in [t_k, t_{n+1}[$, thus reducing this claim to Lemma 145. □

Derivatives of B-Spline Curves

Lemma 146

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \le n$. Then, for the new knot vector $\tau' := (t_1, t_2, \ldots, t_{n+k-1}, t_{n+k})$,

$$\mathcal{P}'(t) = \sum_{i=0}^{n-1} N_{i,k-1,\tau'}(t) q_i \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where

$$q_i := \frac{k}{t_{i+k+1} - t_{i+1}} (p_{i+1} - p_i) \quad \text{for } i \in \{0, 1, \ldots, n-1\}.$$

Sketch of proof: One can show that $N_{i+1,k-1,\tau}(t)$ is equal to $N_{i,k-1,\tau'}(t)$ for all $t \in [t_k, t_{n+1}[$, thus reducing this claim to Lemma 145. □

- Since the first derivative of a B-spline curve is another B-spline curve, one can apply this technique recursively to compute higher-order derivatives.

Corollary 147

A clamped B-spline curve is tangent to the first leg and tangent to the last leg of its control polygon.

Corollary 147

A clamped B-spline curve is tangent to the first leg and tangent to the last leg of its control polygon.

Sketch of proof : Recall that, by Lem. 146, the first derivative of a clamped B-spline curve $\mathcal{P}$ of degree k is a clamped B-spline curve of degree $k-1$ over essentially the same knot vector but with new control points of the form

$$q_i := \frac{k}{t_{i+k+1} - t_{i+1}}(p_{i+1} - p_i) \quad \text{for } i \in \{0, 1, \ldots, n-1\}.$$

Corollary 147

A clamped B-spline curve is tangent to the first leg and tangent to the last leg of its control polygon.

Sketch of proof : Recall that, by Lem. 146, the first derivative of a clamped B-spline curve $\mathcal{P}$ of degree k is a clamped B-spline curve of degree $k-1$ over essentially the same knot vector but with new control points of the form

$$q_i := \frac{k}{t_{i+k+1} - t_{i+1}}(p_{i+1} - p_i) \quad \text{for } i \in \{0, 1, \ldots, n-1\}.$$

Hence, by arguments similar to those used in the proof of Lem. 140, one can show that $\mathcal{P}'(t_k)$ starts in q_0 and, thus, the tangent of $\mathcal{P}$ in the start point $\mathcal{P}(t_k)$ is parallel to $p_1 - p_0$. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Strong Convex Hull Property

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

Strong Convex Hull Property

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

Proof : Lemma 132 tells us that $N_{i-k,k}, N_{i-k+1,k}, \ldots, N_{i-1,k}, N_{i,k}$ are the only B-spline basis functions that can be non-zero over $[t_i, t_{i+1}[$, for $k \leq i \leq n$, while all other basis functions are zero (Lem. 130).

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \le n$. For $i \in \mathbb{N}$ with $k \le i \le n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

Proof : Lemma 132 tells us that $N_{i-k,k}, N_{i-k+1,k}, \ldots, N_{i-1,k}, N_{i,k}$ are the only B-spline basis functions that can be non-zero over $[t_i, t_{i+1}[$, for $k \le i \le n$, while all other basis functions are zero (Lem. 130). Together with Cor. 137, Partition of Unity, we get

$$1 = \sum_{j=0}^{n} N_{j,k}(t) = \sum_{j=i-k}^{i} N_{j,k}(t) \quad \text{for all } t \in [t_i, t_{i+1}[.$$

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \le n$. For $i \in \mathbb{N}$ with $k \le i \le n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

Proof: Lemma 132 tells us that $N_{i-k,k}, N_{i-k+1,k}, \ldots, N_{i-1,k}, N_{i,k}$ are the only B-spline basis functions that can be non-zero over $[t_i, t_{i+1}[$, for $k \le i \le n$, while all other basis functions are zero (Lem. 130). Together with Cor. 137, Partition of Unity, we get

$$1 = \sum_{j=0}^{n} N_{j,k}(t) = \sum_{j=i-k}^{i} N_{j,k}(t) \quad \text{for all } t \in [t_i, t_{i+1}[.$$

Hence,

$$\mathcal{P}(t) = \sum_{j=0}^{n} N_{j,k}(t) p_j = \sum_{j=i-k}^{i} N_{j,k}(t) p_j \quad \text{for all } t \in [t_i, t_{i+1}[$$

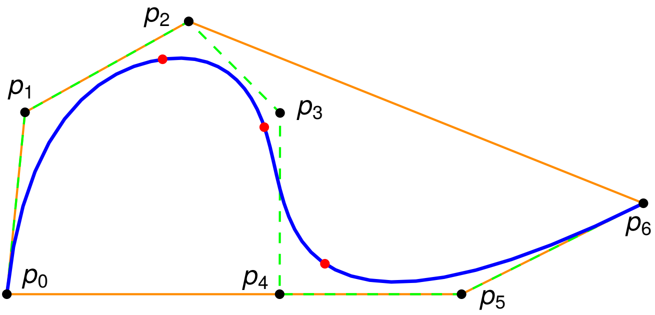
is a convex combination of $p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i$.

Strong Convex Hull Property

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

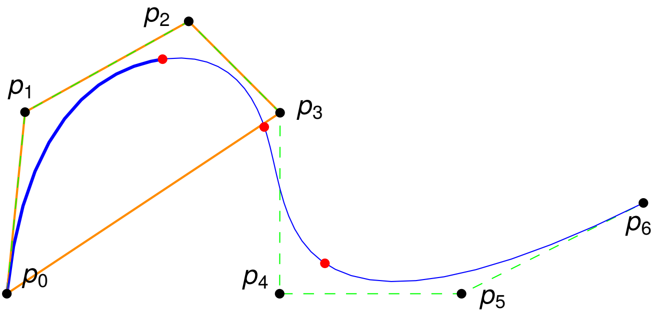


Convex hull of the control polygon of a cubic spline

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \text{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

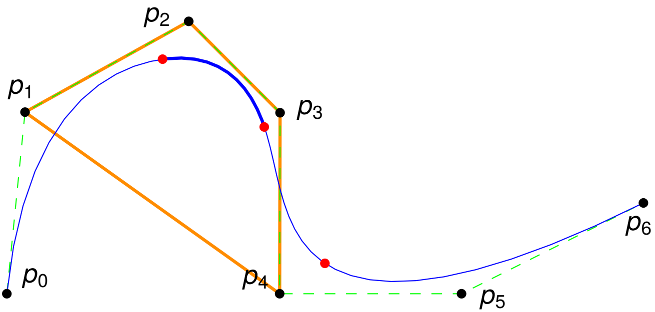


First knot span of a cubic B-spline contained in $\text{CH}(\{p_0, p_1, p_2, p_3\})$.

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \text{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

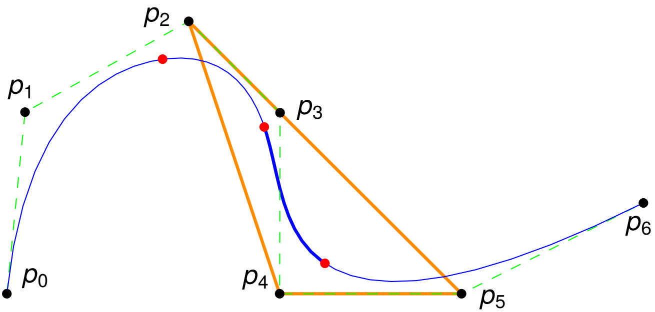


Second knot span of a cubic B-spline contained in $\text{CH}(\{p_1, p_2, p_3, p_4\})$.

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

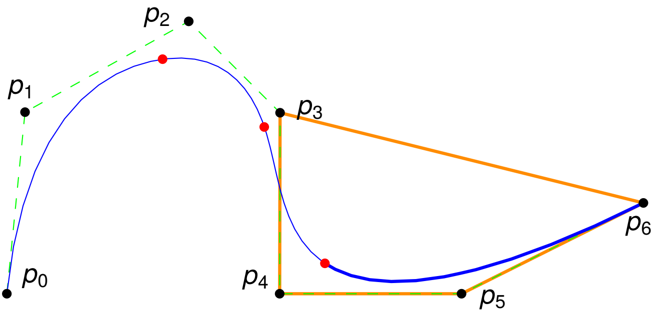


Third knot span of a cubic B-spline contained in $\mathrm{CH}(\{p_2, p_3, p_4, p_5\})$.

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

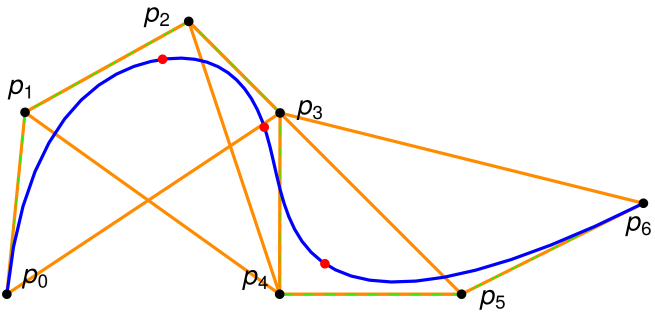


Fourth knot span of a cubic B-spline contained in $\mathrm{CH}(\{p_3, p_4, p_5, p_6\})$.

Lemma 148 (Strong convex hull property)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{P}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$



Union of individual hulls yields tighter enclosing shape.

Lemma 149 (Local control)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then the B-spline curve $\mathcal{P}$ restricted to $[t_i, t_{i+1}[$ depends only on the positions of $p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 149 (Local control)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then the B-spline curve $\mathcal{P}$ restricted to $[t_i, t_{i+1}[$ depends only on the positions of $p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i$.

Proof : By Lem. 132, and as in the proof of Lem. 148,

$$\mathcal{P}|_{[t_i,t_{i+1}[}(t) = \sum_{j=i-k}^{i} N_{j,k}(t)p_j.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Local Control and Modification

Lemma 149 (Local control)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then the B-spline curve $\mathcal{P}$ restricted to $[t_i, t_{i+1}[$ depends only on the positions of $p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i$.

Proof: By Lem. 132, and as in the proof of Lem. 148,

$$\mathcal{P}|_{[t_i,t_{i+1}[}(t) = \sum_{j=i-k}^{i} N_{j,k}(t)p_j. \qquad \square$$

Lemma 150 (Local modification scheme)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then a modification of the position of p_i changes $\mathcal{P}$ only in the parameter interval $[t_i, t_{i+k+1}[$, for $i \in \{0, 1, \ldots, n\}$.

Lemma 149 (Local control)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then the B-spline curve $\mathcal{P}$ restricted to $[t_i, t_{i+1}[$ depends only on the positions of $p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i$.

Proof: By Lem. 132, and as in the proof of Lem. 148,

$$\mathcal{P}|_{[t_i, t_{i+1}[}(t) = \sum_{j=i-k}^{i} N_{j,k}(t) p_j. \qquad \square$$

Lemma 150 (Local modification scheme)

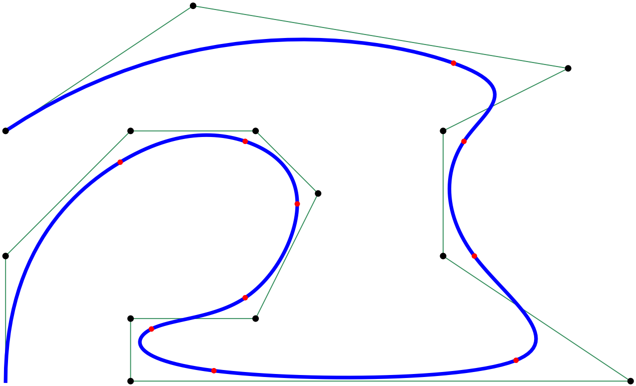
Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Then a modification of the position of p_i changes $\mathcal{P}$ only in the parameter interval $[t_i, t_{i+k+1}[$, for $i \in \{0, 1, \ldots, n\}$.

Proof: The Local Support Lemma 129 tells us that

$$N_{i,k}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

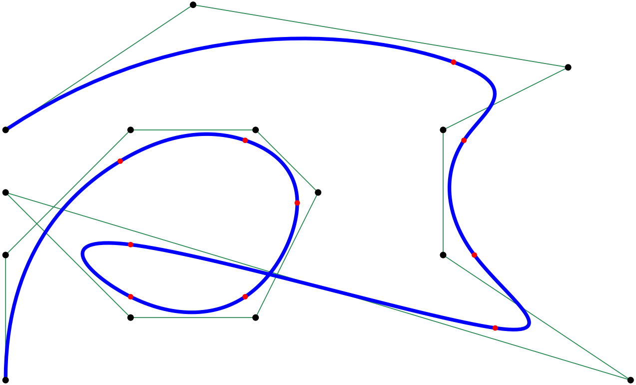
Local Control and Modification

- Clamped uniform B-spline of degree three with knot vector
$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11, 11)$$
for a control polygon with 14 vertices:
$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix} \right\}$$



Local Control and Modification

- Clamped uniform B-spline of degree three with knot vector
$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11, 11)$$
for a control polygon with 14 vertices:
$$\left\{\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1\\3\end{pmatrix}, \begin{pmatrix}3\\5\end{pmatrix}, \begin{pmatrix}5\\5\end{pmatrix}, \begin{pmatrix}6\\4\end{pmatrix}, \begin{pmatrix}5\\2\end{pmatrix}, \begin{pmatrix}3\\2\end{pmatrix}, \begin{pmatrix}1\\4\end{pmatrix}, \begin{pmatrix}11\\1\end{pmatrix}, \begin{pmatrix}8\\3\end{pmatrix}, \begin{pmatrix}8\\5\end{pmatrix}, \begin{pmatrix}10\\6\end{pmatrix}, \begin{pmatrix}4\\7\end{pmatrix}, \begin{pmatrix}1\\5\end{pmatrix}\right\}$$



Multiple Control Points

Lemma 151 (Multiple control points)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 151 (Multiple control points)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$.

1. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ coincide, i.e., if $p_{i-k+1} = p_{i-k+2} = \ldots = p_i$ then $\mathcal{P}$ contains p_i and is tangent to the legs $\overline{p_{i-k}p_{i-k+1}}$ and $\overline{p_ip_{i+1}}$ of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 151 (Multiple control points)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$.

1. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ coincide, i.e., if $p_{i-k+1} = p_{i-k+2} = \ldots = p_i$ then $\mathcal{P}$ contains p_i and is tangent to the legs $\overline{p_{i-k}p_{i-k+1}}$ and $\overline{p_i p_{i+1}}$ of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.
2. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ are collinear then $\mathcal{P}$ touches a leg of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.

Lemma 151 (Multiple control points)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$.

1. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ coincide, i.e., if $p_{i-k+1} = p_{i-k+2} = \ldots = p_i$ then $\mathcal{P}$ contains p_i and is tangent to the legs $\overline{p_{i-k}p_{i-k+1}}$ and $\overline{p_i p_{i+1}}$ of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.
2. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ are collinear then $\mathcal{P}$ touches a leg of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.
3. If $k+1$ control points $p_{i-k}, p_{i-k+1}, \ldots, p_i$ are collinear then $\mathcal{P}$ coincides with a leg of the control polygon, for $i \in \mathbb{N}$ with $k < i < n$.

Lemma 151 (Multiple control points)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$.

1. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ coincide, i.e., if $p_{i-k+1} = p_{i-k+2} = \ldots = p_i$ then $\mathcal{P}$ contains p_i and is tangent to the legs $\overline{p_{i-k}p_{i-k+1}}$ and $\overline{p_i p_{i+1}}$ of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.
2. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ are collinear then $\mathcal{P}$ touches a leg of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.
3. If $k+1$ control points $p_{i-k}, p_{i-k+1}, \ldots, p_i$ are collinear then $\mathcal{P}$ coincides with a leg of the control polygon, for $i \in \mathbb{N}$ with $k < i < n$.

Sketch of proof: This is a consequence of the Local Control Lemma 149 and of the Strong Convex Hull Property (Lem. 148). □

Multiple Control Points

Lemma 151 (Multiple control points)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$.

1. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ coincide, i.e., if $p_{i-k+1} = p_{i-k+2} = \ldots = p_i$ then $\mathcal{P}$ contains p_i and is tangent to the legs $\overline{p_{i-k}p_{i-k+1}}$ and $\overline{p_i p_{i+1}}$ of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.
2. If k control points $p_{i-k+1}, p_{i-k+2}, \ldots, p_i$ are collinear then $\mathcal{P}$ touches a leg of the control polygon, for $i \in \mathbb{N}$ with $k \leq i < n$.
3. If $k+1$ control points $p_{i-k}, p_{i-k+1}, \ldots, p_i$ are collinear then $\mathcal{P}$ coincides with a leg of the control polygon, for $i \in \mathbb{N}$ with $k < i < n$.

Sketch of proof : This is a consequence of the Local Control Lemma 149 and of the Strong Convex Hull Property (Lem. 148). □

- Note that this implies that a degree-k B-spline $\mathcal{P}$ starts at p_0 if $p_0 = p_1 = \ldots = p_{k-1}$.

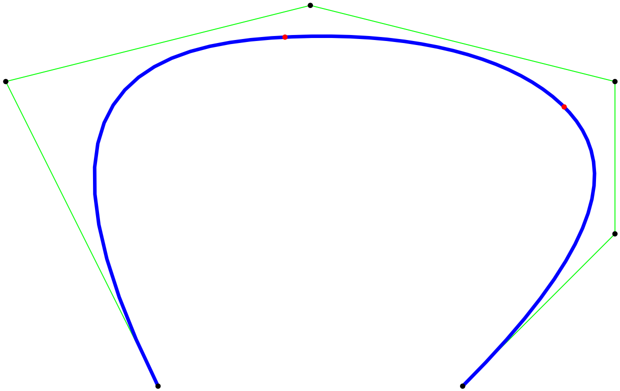
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Multiple Control Points

Clamped cubic B-spline with control points

$$\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\4\end{pmatrix}, \begin{pmatrix}4\\5\end{pmatrix}, \begin{pmatrix}8\\4\end{pmatrix}, \begin{pmatrix}8\\2\end{pmatrix}, \begin{pmatrix}6\\0\end{pmatrix}$$

and uniform knot vector $(0, 0, 0, 0, 1, 2, 3, 3, 3, 3)$:

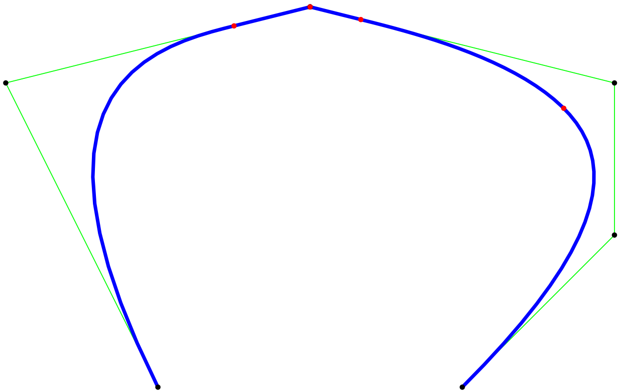


Multiple Control Points

Clamped cubic B-spline with control points

$$\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\4\end{pmatrix}, \begin{pmatrix}4\\5\end{pmatrix}, \begin{pmatrix}4\\5\end{pmatrix}, \begin{pmatrix}4\\5\end{pmatrix}, \begin{pmatrix}8\\4\end{pmatrix}, \begin{pmatrix}8\\2\end{pmatrix}, \begin{pmatrix}6\\0\end{pmatrix}$$

and uniform knot vector $(0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5)$:

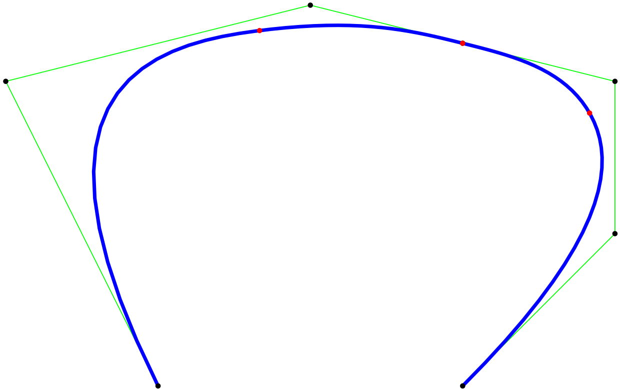


Multiple Control Points

Clamped cubic B-spline with control points

$$\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\4\end{pmatrix}, \begin{pmatrix}4\\5\end{pmatrix}, \begin{pmatrix}6\\\frac{9}{2}\end{pmatrix}, \begin{pmatrix}8\\4\end{pmatrix}, \begin{pmatrix}8\\2\end{pmatrix}, \begin{pmatrix}6\\0\end{pmatrix}$$

and uniform knot vector $(0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$:

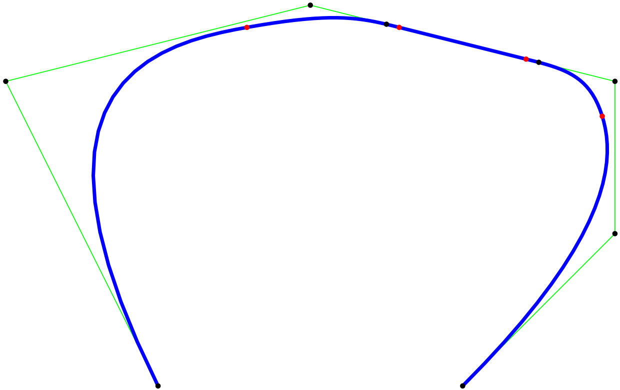


Multiple Control Points

Clamped cubic B-spline with control points

$$\begin{pmatrix}2\\0\end{pmatrix},\begin{pmatrix}0\\4\end{pmatrix},\begin{pmatrix}4\\5\end{pmatrix},\begin{pmatrix}5\\\frac{19}{4}\end{pmatrix},\begin{pmatrix}7\\\frac{17}{4}\end{pmatrix},\begin{pmatrix}8\\4\end{pmatrix},\begin{pmatrix}8\\2\end{pmatrix},\begin{pmatrix}6\\0\end{pmatrix}$$

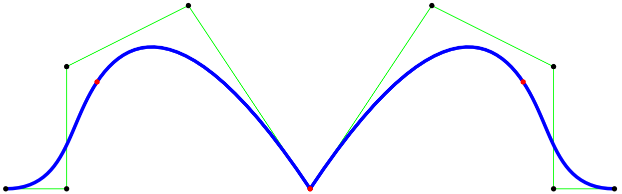
and uniform knot vector $(0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5)$:



Multiple Knots

Lemma 152 (Multiple knots)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Let $i \in \mathbb{N}$ with $k+1 \leq i \leq n-k$. If t_i is a knot of multiplicity k, i.e., if $t_i = t_{i+1} = \ldots = t_{i+k-1}$ then $\mathcal{P}(t_i) = p_{i-1}$ and $\mathcal{P}$ is tangent to the legs $\overline{p_{i-2}p_{i-1}}$ and $\overline{p_{i-1}p_i}$ of the control polygon.



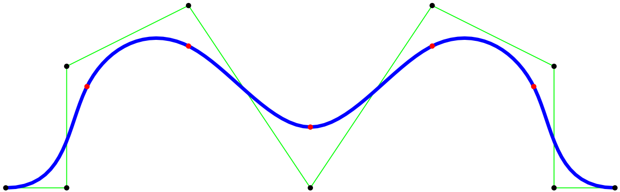
Multiple Knots

- Clamped uniform B-spline of degree three for a control polygon with nine vertices:

$$\left\{ \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 6, 6, 6)$$



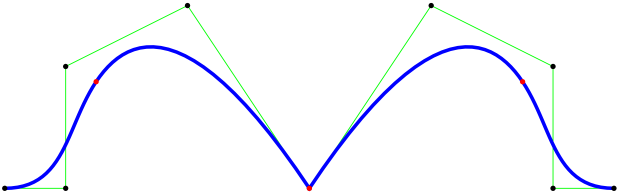
Multiple Knots

- Clamped uniform B-spline of degree three for a control polygon with nine vertices:

$$\left\{ \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix}, \begin{pmatrix} 9 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 2, 2, 3, 4, 4, 4, 4)$$



Motivation for de Boor's Algorithm

- Can we express $\mathcal{P}(t)$ in terms of $N_{i,0}(t)$?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation for de Boor's Algorithm

- Can we express $\mathcal{P}(t)$ in terms of $N_{i,0}(t)$?
- We exploit the recursive Definition 128 of $N_{i,k}(t)$ in order to determine $\mathcal{P}(t)$ in terms of $N_{i,k-1}(t)$, recalling that $t \in [t_k, t_{n+1}[$.

$$\mathcal{P}(t) = \sum_{i=0}^{n} N_{i,k}(t) p_i = \sum_{i=0}^{n} \left(\frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \right) p_i$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation for de Boor's Algorithm

- Can we express $\mathcal{P}(t)$ in terms of $N_{i,0}(t)$?
- We exploit the recursive Definition 128 of $N_{i,k}(t)$ in order to determine $\mathcal{P}(t)$ in terms of $N_{i,k-1}(t)$, recalling that $t \in [t_k, t_{n+1}[$.

$$\mathcal{P}(t) = \sum_{i=0}^{n} N_{i,k}(t) p_i = \sum_{i=0}^{n} \left(\frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \right) p_i$$

$$= \sum_{i=0}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) p_i + \sum_{i=0}^{n} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) p_i$$

Motivation for de Boor's Algorithm

- Can we express $\mathcal{P}(t)$ in terms of $N_{i,0}(t)$?
- We exploit the recursive Definition 128 of $N_{i,k}(t)$ in order to determine $\mathcal{P}(t)$ in terms of $N_{i,k-1}(t)$, recalling that $t \in [t_k, t_{n+1}[$.

$$
\begin{aligned}
\mathcal{P}(t) &= \sum_{i=0}^{n} N_{i,k}(t) p_i = \sum_{i=0}^{n} \left(\frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \right) p_i \\
&= \sum_{i=0}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) p_i + \sum_{i=0}^{n} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) p_i \\
&= \frac{t - t_0}{t_k - t_0} N_{0,k-1}(t) p_0 + \sum_{i=1}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) p_i + \\
&\quad \frac{t_{n+k+1} - t}{t_{n+k+1} - t_{n+1}} N_{n+1,k-1}(t) p_n + \sum_{i=0}^{n-1} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) p_i
\end{aligned}
$$

Motivation for de Boor's Algorithm

- Can we express $\mathcal{P}(t)$ in terms of $N_{i,0}(t)$?
- We exploit the recursive Definition 128 of $N_{i,k}(t)$ in order to determine $\mathcal{P}(t)$ in terms of $N_{i,k-1}(t)$, recalling that $t \in [t_k, t_{n+1}[$.

$$
\begin{aligned}
\mathcal{P}(t) &= \sum_{i=0}^{n} N_{i,k}(t)p_i = \sum_{i=0}^{n} \left(\frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \right) p_i \\
&= \sum_{i=0}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t)p_i + \sum_{i=0}^{n} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t)p_i \\
&= \frac{t - t_0}{t_k - t_0} N_{0,k-1}(t)p_0 + \sum_{i=1}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t)p_i + \\
&\quad \frac{t_{n+k+1} - t}{t_{n+k+1} - t_{n+1}} N_{n+1,k-1}(t)p_n + \sum_{i=0}^{n-1} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t)p_i \\
&\stackrel{\star}{=} \sum_{i=1}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t)p_i + \sum_{i=1}^{n} \frac{t_{i+k} - t}{t_{i+k} - t_i} N_{i,k-1}(t)p_{i-1}
\end{aligned}
$$

Motivation for de Boor's Algorithm

- Can we express $\mathcal{P}(t)$ in terms of $N_{i,0}(t)$?
- We exploit the recursive Definition 128 of $N_{i,k}(t)$ in order to determine $\mathcal{P}(t)$ in terms of $N_{i,k-1}(t)$, recalling that $t \in [t_k, t_{n+1}[$.

$$\mathcal{P}(t) = \sum_{i=0}^{n} N_{i,k}(t) p_i = \sum_{i=0}^{n} \left(\frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \right) p_i$$

$$= \sum_{i=0}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) p_i + \sum_{i=0}^{n} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) p_i$$

$$= \frac{t - t_0}{t_k - t_0} N_{0,k-1}(t) p_0 + \sum_{i=1}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) p_i +$$

$$\frac{t_{n+k+1} - t}{t_{n+k+1} - t_{n+1}} N_{n+1,k-1}(t) p_n + \sum_{i=0}^{n-1} \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) p_i$$

$$\stackrel{\star}{=} \sum_{i=1}^{n} \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) p_i + \sum_{i=1}^{n} \frac{t_{i+k} - t}{t_{i+k} - t_i} N_{i,k-1}(t) p_{i-1}$$

$$= \sum_{i=1}^{n} N_{i,k-1}(t) \left(\frac{t_{i+k} - t}{t_{i+k} - t_i} p_{i-1} + \frac{t - t_i}{t_{i+k} - t_i} p_i \right)$$

Motivation for de Boor's Algorithm

- Can we express $\mathcal{P}(t)$ in terms of $N_{i,0}(t)$?
- We exploit the recursive Definition 128 of $N_{i,k}(t)$ in order to determine $\mathcal{P}(t)$ in terms of $N_{i,k-1}(t)$, recalling that $t \in [t_k, t_{n+1}[$.

$$
\begin{aligned}
\mathcal{P}(t) &= \sum_{i=0}^{n} N_{i,k}(t)p_i = \sum_{i=0}^{n}\left(\frac{t-t_i}{t_{i+k}-t_i}N_{i,k-1}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t)\right)p_i \\
&= \sum_{i=0}^{n}\frac{t-t_i}{t_{i+k}-t_i}N_{i,k-1}(t)p_i + \sum_{i=0}^{n}\frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t)p_i \\
&= \frac{t-t_0}{t_k-t_0}N_{0,k-1}(t)p_0 + \sum_{i=1}^{n}\frac{t-t_i}{t_{i+k}-t_i}N_{i,k-1}(t)p_i + \\
&\quad \frac{t_{n+k+1}-t}{t_{n+k+1}-t_{n+1}}N_{n+1,k-1}(t)p_n + \sum_{i=0}^{n-1}\frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t)p_i \\
&\stackrel{\star}{=} \sum_{i=1}^{n}\frac{t-t_i}{t_{i+k}-t_i}N_{i,k-1}(t)p_i + \sum_{i=1}^{n}\frac{t_{i+k}-t}{t_{i+k}-t_i}N_{i,k-1}(t)p_{i-1} \\
&= \sum_{i=1}^{n}N_{i,k-1}(t)\left(\frac{t_{i+k}-t}{t_{i+k}-t_i}p_{i-1} + \frac{t-t_i}{t_{i+k}-t_i}p_i\right) =: \sum_{i=1}^{n}N_{i,k-1}(t)p_{i,1}(t)
\end{aligned}
$$

Motivation for de Boor's Algorithm

- Equality at $\star$ holds since each basis function $N_{i,k}$ is non-zero only over $[t_i, t_{i+k+1}[$ (Local Support Lem. 129):

$$N_{0,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_0, t_k[\quad \text{and} \quad N_{n+1,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_{n+1}, t_{n+k+1}[$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation for de Boor's Algorithm

- Equality at $\star$ holds since each basis function $N_{i,k}$ is non-zero only over $[t_i, t_{i+k+1}[$ (Local Support Lem. 129):

$$N_{0,k-1,\tau}(t) = 0 \ \text{ for } \ t \notin [t_0, t_k[\quad \text{and} \quad N_{n+1,k-1,\tau}(t) = 0 \ \text{ for } \ t \notin [t_{n+1}, t_{n+k+1}[$$

- For $1 \le i \le n$, we have

$$p_{i,1}(t) := (1 - \alpha_{i,1})\, p_{i-1} + \alpha_{i,1}\, p_i \qquad \text{with } \alpha_{i,1} := \frac{t - t_i}{t_{i+k} - t_i},$$

thus expressing $\mathcal{P}(t)$ in terms of basis functions of degree $k-1$ and modified (parameter-dependent!) new control points.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Motivation for de Boor's Algorithm

- Equality at $\star$ holds since each basis function $N_{i,k}$ is non-zero only over $[t_i, t_{i+k+1}[$ (Local Support Lem. 129):

$$N_{0,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_0, t_k[\quad \text{and} \quad N_{n+1,k-1,\tau}(t) = 0 \;\text{ for }\; t \notin [t_{n+1}, t_{n+k+1}[$$

- For $1 \leq i \leq n$, we have

$$p_{i,1}(t) := (1 - \alpha_{i,1})\, p_{i-1} + \alpha_{i,1}\, p_i \qquad \text{with } \alpha_{i,1} := \frac{t - t_i}{t_{i+k} - t_i},$$

thus expressing $\mathcal{P}(t)$ in terms of basis functions of degree $k-1$ and modified (parameter-dependent!) new control points.

- Repeating this process yields

$$\mathcal{P}(t) = \sum_{i=2}^{n} N_{i,k-2}(t) p_{i,2}(t),$$

where, for $2 \leq i \leq n$,

$$p_{i,2}(t) := (1 - \alpha_{i,2})\, p_{i-1,1}(t) + \alpha_{i,2}\, p_{i,1}(t) \qquad \text{with } \alpha_{i,2} := \frac{t - t_i}{t_{i+k-1} - t_i},$$

Theorem 153 (de Boor's algorithm)

Let $\mathcal{P}$ be a B-spline curve of degree k with control points $p_0, p_1, \ldots, p_n$ and knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$. If we define

$$p_{i,j}(t) := \begin{cases} p_i & \text{if } j = 0, \\ (1 - \alpha_{i,j})\, p_{i-1,j-1}(t) + \alpha_{i,j}\, p_{i,j-1}(t) & \text{if } j > 0, \end{cases}$$

where

$$\alpha_{i,j} := \frac{t - t_i}{t_{i+k+1-j} - t_i},$$

then

$$\mathcal{P}(t) = \sum_{i=k}^{n} N_{i,0}(t) p_{i,k}(t) \qquad \text{for } t \in [t_k, t_{n+1}[.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

De Boor's Algorithm

Theorem 153 (de Boor's algorithm)

Let $\mathcal{P}$ be a B-spline curve of degree k with control points $p_0, p_1, \ldots, p_n$ and knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$. If we define

$$p_{i,j}(t) := \begin{cases} p_i & \text{if } j = 0, \\ (1 - \alpha_{i,j})\, p_{i-1,j-1}(t) + \alpha_{i,j}\, p_{i,j-1}(t) & \text{if } j > 0, \end{cases}$$

where

$$\alpha_{i,j} := \frac{t - t_i}{t_{i+k+1-j} - t_i},$$

then

$$\mathcal{P}(t) = \sum_{i=k}^{n} N_{i,0}(t) p_{i,k}(t) \qquad \text{for } t \in [t_k, t_{n+1}[.$$

Corollary 154

Let $\mathcal{P}$ be a B-spline curve of degree k with control points $p_0, p_1, \ldots, p_n$ and knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$. If $t \in [t_i, t_{i+1}[$, for $i \in \{k, k+1, \ldots, n\}$, then $\mathcal{P}(t) = p_{i,k}(t)$.

Sample Run of de Boor's Algorithm

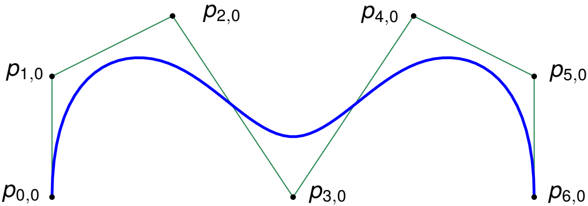
- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- B-spline curve with $p_{i,0}(0.7)$, with $0.7 \in [t_3, t_4[$:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$



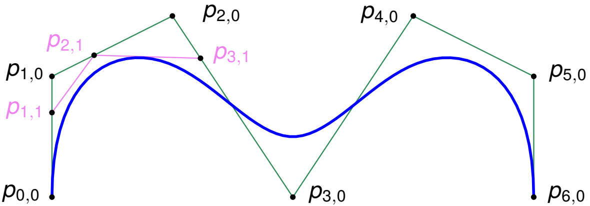
- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- B-spline curve with $p_{i,1}(0.7)$, with $0.7 \in [t_3, t_4[$:

$$\left\{ \begin{pmatrix} 0. \\ 1.4 \end{pmatrix}, \begin{pmatrix} 0.7 \\ 2.35 \end{pmatrix}, \begin{pmatrix} 2.4667 \\ 2.3 \end{pmatrix} \right\}$$



Sample Run of de Boor's Algorithm

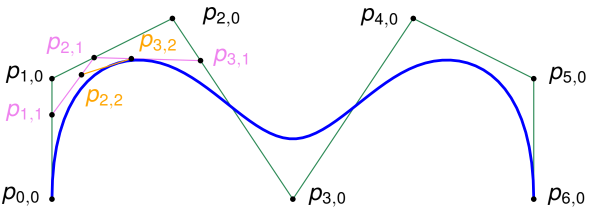
- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- B-spline curve with $p_{i,2}(0.7)$, with $0.7 \in [t_3, t_4[$:

$$\left\{ \begin{pmatrix} 0.49 \\ 2.065 \end{pmatrix}, \begin{pmatrix} 1.3183 \\ 2.3325 \end{pmatrix} \right\}$$



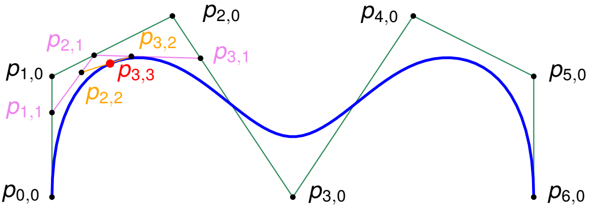
- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- B-spline curve with $p_{i,3}(0.7)$, with $0.7 \in [t_3, t_4[$:

$$\left\{ \begin{pmatrix} 1.0698 \\ 2.2523 \end{pmatrix} \right\} = \{\mathcal{P}(0.7)\}$$

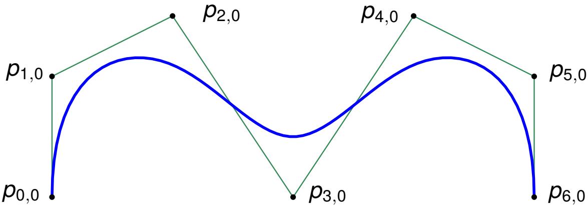


De Boor's Algorithm for Subdividing a B-Spline Curve

- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.



De Boor's Algorithm for Subdividing a B-Spline Curve

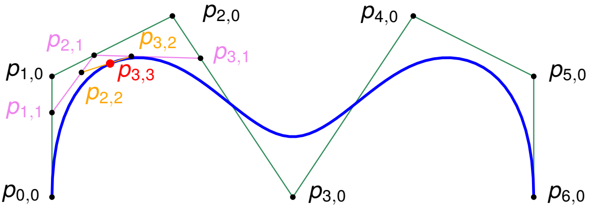
- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- New control polygons for $t^\star := 0.7$:

$$(p_{0,0}, p_{1,1}, p_{2,2}, p_{3,3}) \quad \text{and} \quad (p_{3,3}, p_{3,2}, p_{3,1}, p_{3,0}, p_{4,0}, p_{5,0}, p_{6,0})$$



De Boor's Algorithm for Subdividing a B-Spline Curve

- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

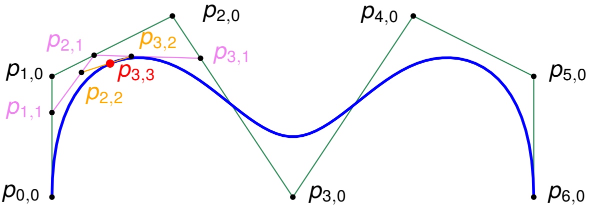
  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- New control polygons for $t^\star := 0.7$:

$$(p_{0,0}, p_{1,1}, p_{2,2}, p_{3,3}) \quad \text{and} \quad (p_{3,3}, p_{3,2}, p_{3,1}, p_{3,0}, p_{4,0}, p_{5,0}, p_{6,0})$$

- New knot vectors for $t^\star := 0.7$:

$$(0, 0, 0, 0, 0.7, 0.7, 0.7, 0.7) \quad \text{and} \quad (0.7, 0.7, 0.7, 0.7, 1, 2, 3, 4, 4, 4, 4)$$



Definition 155

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For some $t^\star \in [t_i, t_{i+1}[$, with $i \in \{k, \ldots, n\}$, we define two new knot vectors $\tau^\star, \tau^{\star\star}$ and two new control polygons $P^\star, P^{\star\star}$ as follows:

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

De Boor's Algorithm for Subdividing a B-Spline Curve

Definition 155

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For some $t^\star \in [t_i, t_{i+1}[$, with $i \in \{k, \ldots, n\}$, we define two new knot vectors $\tau^\star, \tau^{\star\star}$ and two new control polygons $P^\star, P^{\star\star}$ as follows:
If $t^\star \neq t_i$ then $m := i$ else $m := i - 1$.

$$\tau^\star := (t_0, t_1, \ldots, t_m, \underbrace{t^\star, \ldots, t^\star}_{(k+1) \text{ times}}) \quad \text{and} \quad \tau^{\star\star} := (\underbrace{t^\star, \ldots, t^\star}_{(k+1) \text{ times}}, t_{m+1}, \ldots, t_{n+k+1}),$$

© M. Held (Univ. Salzburg)

De Boor's Algorithm for Subdividing a B-Spline Curve

Definition 155

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. For some $t^\star \in [t_i, t_{i+1}[$, with $i \in \{k, \ldots, n\}$, we define two new knot vectors $\tau^\star, \tau^{\star\star}$ and two new control polygons $P^\star, P^{\star\star}$ as follows:
If $t^\star \neq t_i$ then $m := i$ else $m := i - 1$.

$$\tau^\star := (t_0, t_1, \ldots, t_m, \underbrace{t^\star, \ldots, t^\star}_{(k+1) \text{ times}}) \quad \text{and} \quad \tau^{\star\star} := (\underbrace{t^\star, \ldots, t^\star}_{(k+1) \text{ times}}, t_{m+1}, \ldots, t_{n+k+1}),$$

$$P^\star(t^\star) := (p_{0,0}(t^\star), p_{1,0}(t^\star), \ldots, p_{m-k,0}(t^\star), p_{1,1}(t^\star), p_{2,2}(t^\star), \ldots, p_{k,k}(t^\star)),$$

$$P^{\star\star}(t^\star) := (p_{k,k}(t^\star), p_{k,k-1}(t^\star), \ldots, p_{k,1}(t^\star), p_{m,0}(t^\star), p_{m+1,0}(t^\star), \ldots, p_{n,0}(t^\star)),$$

where the new control points $p_{i,j}(t^\star)$ are obtained by de Boor's algorithm (Thm. 153).

Lemma 156

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. (Hence, $t_0 = t_1 = \ldots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \ldots = t_{n+k+1}$.) For some $t^\star \in [t_i, t_{i+1}[$, with $i \in \{k, \ldots, n\}$, we define two new knot vectors $\tau^\star, \tau^{\star\star}$ and two new control polygons $P^\star, P^{\star\star}$ as in Def. 155.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 156

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. (Hence, $t_0 = t_1 = \ldots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \ldots = t_{n+k+1}$.) For some $t^\star \in [t_i, t_{i+1}[$, with $i \in \{k, \ldots, n\}$, we define two new knot vectors $\tau^\star, \tau^{\star\star}$ and two new control polygons $P^\star, P^{\star\star}$ as in Def. 155. Then we get two new B-spline curves $\mathcal{P}^\star$ and $\mathcal{P}^{\star\star}$ of degree k with control polygon $P^\star$ ($P^{\star\star}$, resp.) and knot vector $\tau^\star$ ($\tau^{\star\star}$, resp.) that join in a tangent-continuous way at point $p_{kk}(t^\star) = \mathcal{P}(t^\star)$, such that

$$\mathcal{P}^\star = \mathcal{P}|_{[t_k, t^\star[} \quad \text{and} \quad \mathcal{P}^{\star\star} = \mathcal{P}|_{[t^\star, t_{n+1}[}.$$

Corollary 157

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. (Hence, $t_0 = t_1 = \ldots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \ldots = t_{n+k+1}$.) Subdividing $\mathcal{P}$ at the knot values $\{t_{k+1}, t_{k+2}, \ldots, t_{n-1}, t_n\}$, as outlined in Def. 155, splits $\mathcal{P}$ into $n-k+1$ Bézier curves of degree k.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Corollary 157

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. (Hence, $t_0 = t_1 = \ldots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \ldots = t_{n+k+1}$.) Subdividing $\mathcal{P}$ at the knot values $\{t_{k+1}, t_{k+2}, \ldots, t_{n-1}, t_n\}$, as outlined in Def. 155, splits $\mathcal{P}$ into $n-k+1$ Bézier curves of degree k.

Sketch of proof : Lemma 156 ensures that each of the resulting curves is a B-spline curve of degree k, where the m-th curve is defined over $[t_{k+m}, t_{k+m+1}[$, for $m \in \{0, 1, \ldots, n-k\}$.

Corollary 157

Let $\mathcal{P}$ be a clamped B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \dots, p_n$ and the clamped knot vector $\tau := (t_0, t_1, \dots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. (Hence, $t_0 = t_1 = \dots = t_k < t_{k+1}$ and $t_n < t_{n+1} = \dots = t_{n+k+1}$.) Subdividing $\mathcal{P}$ at the knot values $\{t_{k+1}, t_{k+2}, \dots, t_{n-1}, t_n\}$, as outlined in Def. 155, splits $\mathcal{P}$ into $n-k+1$ Bézier curves of degree k.

Sketch of proof: Lemma 156 ensures that each of the resulting curves is a B-spline curve of degree k, where the m-th curve is defined over $[t_{k+m}, t_{k+m+1}[$, for $m \in \{0, 1, \dots, n-k\}$. Each curve has knot vectors of length $2k+2$, with start and end knots of multiplicity $k+1$ but no interior knots. After mapping $[t_{k+m}, t_{k+m+1}[$ to $[0, 1[$ we can apply Lem. 144 and conclude that the resulting B-spline curve is a Bézier curve of degree k. □

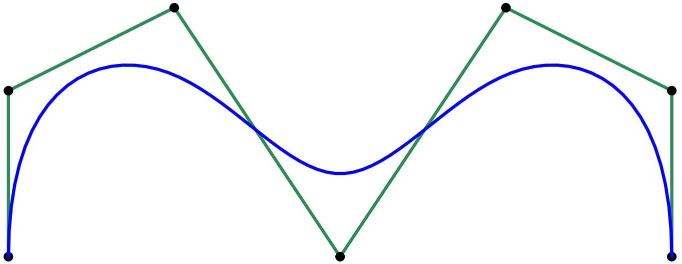
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

De Boor's Algorithm for Splitting a B-Spline Curve into Bézier Segments

- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

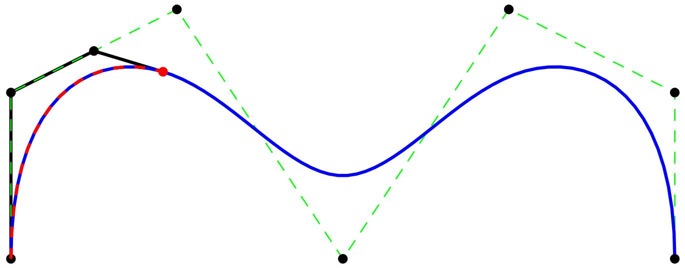
  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.
- Original B-spline curve.



- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.
- First Bézier segment over $[0, 1]$.

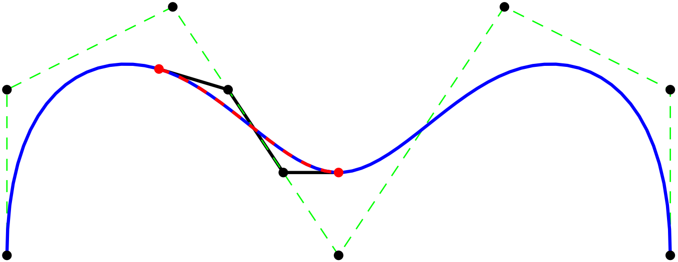


De Boor's Algorithm for Splitting a B-Spline Curve into Bézier Segments

- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.
- Second Bézier segment over $[1, 2]$.



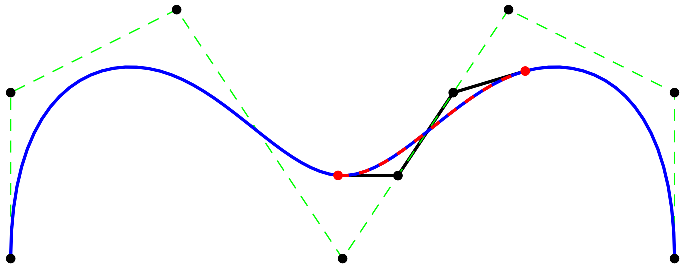
De Boor's Algorithm for Splitting a B-Spline Curve into Bézier Segments

- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- Third Bézier segment over $[2, 3]$.

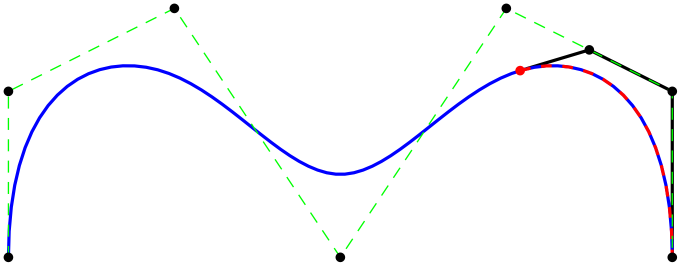


De Boor's Algorithm for Splitting a B-Spline Curve into Bézier Segments

- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.
- Fourth Bézier segment over $[3, 4]$.



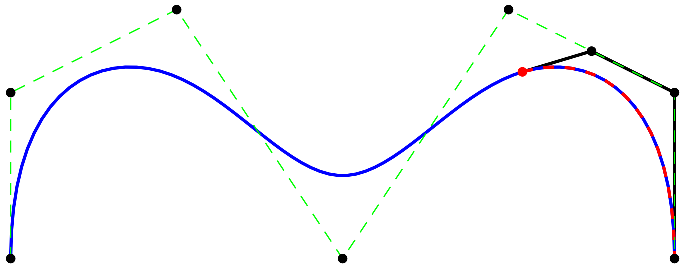
De Boor's Algorithm for Splitting a B-Spline Curve into Bézier Segments

- Clamped uniform B-spline of degree three for seven control points:

$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector with eleven knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$.

- Fourth Bézier segment over $[3, 4]$.



- Note that the number of knots increased drasticallly!

Knot Insertion

- Suppose that we would like to insert a new knot $t^\star \in [t_j, t_{j+1}[$, for some $j \in \{k, k+1, \ldots, n\}$, into the knot vector

$$\tau := (t_0, t_1, \ldots, t_j, t_{j+1}, \ldots, t_{n+k+1}),$$

thus transforming τ into a knot vector

$$\tau^\star := (t_0, t_1, \ldots, t_j, t^\star, t_{j+1}, \ldots, t_{n+k+1}).$$

Knot Insertion

- Suppose that we would like to insert a new knot $t^\star \in [t_j, t_{j+1}[$, for some $j \in \{k, k+1, \ldots, n\}$, into the knot vector

  $$\tau := (t_0, t_1, \ldots, t_j, t_{j+1}, \ldots, t_{n+k+1}),$$

  thus transforming τ into a knot vector

  $$\tau^\star := (t_0, t_1, \ldots, t_j, t^\star, t_{j+1}, \ldots, t_{n+k+1}).$$

- The fundamental equality $m = n + k + 1$, with $m + 1$ denoting the number of knots, tells us that we will have to either increase the number n of control points by one or to increase the degree k of the curve by one.

Knot Insertion

- Suppose that we would like to insert a new knot $t^\star \in [t_j, t_{j+1}[$, for some $j \in \{k, k+1, \ldots, n\}$, into the knot vector

  $$\tau := (t_0, t_1, \ldots, t_j, t_{j+1}, \ldots, t_{n+k+1}),$$

  thus transforming τ into a knot vector

  $$\tau^\star := (t_0, t_1, \ldots, t_j, t^\star, t_{j+1}, \ldots, t_{n+k+1}).$$

- The fundamental equality $m = n + k + 1$, with $m + 1$ denoting the number of knots, tells us that we will have to either increase the number n of control points by one or to increase the degree k of the curve by one.
- Since an increase of the degree would change the shape of the B-spline globally, we opt for increasing the number of control points (and modifying some of them).
- How can we modify the control points such that the shape of the curve is preserved?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Insertion

Lemma 158 (Boehm 1980)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Let $t^\star \in [t_j, t_{j+1}[$, for some $j \in \{k, k+1, \ldots, n\}$, and define a knot vector $\tau^\star$ as $\tau^\star := (t_0, t_1, \ldots, t_j, t^\star, t_{j+1}, \ldots, t_{n+k+1})$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 158 (Boehm 1980)

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. Let $t^\star \in [t_j, t_{j+1}[$, for some $j \in \{k, k+1, \ldots, n\}$, and define a knot vector $\tau^\star$ as $\tau^\star := (t_0, t_1, \ldots, t_j, t^\star, t_{j+1}, \ldots, t_{n+k+1})$. Then we have

$$\mathcal{P}(t) = \sum_{i=0}^{n} N_{i,k,\tau}(t) p_i = \sum_{i=0}^{n+1} N_{i,k,\tau^\star}(t) p_i^\star =: \mathcal{P}^\star(t) \quad \text{for all } t \in [t_k, t_{n+1}[$$

if, for $0 \leq i \leq n+1$,

$$p_i^\star := \begin{cases} p_i & \text{if } i \leq j-k \\ (1-\alpha_i)p_{i-1} + \alpha_i p_i & \text{if } j-k+1 \leq i \leq j \\ p_{i-1} & \text{if } i \geq j+1 \end{cases}$$

and

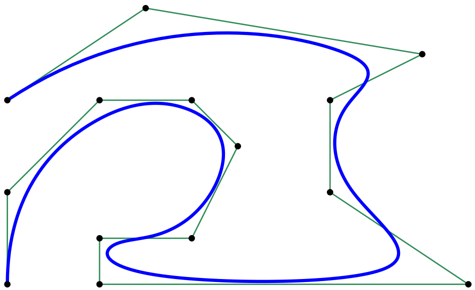
$$\alpha_i := \frac{t^\star - t_i}{t_{i+k} - t_i} \qquad \text{for } i \in \{j-k+1, \ldots, j\}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Insertion: Sample

- Clamped uniform B-spline of degree three for 14 control points and knot vector with 18 knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11, 11)$.

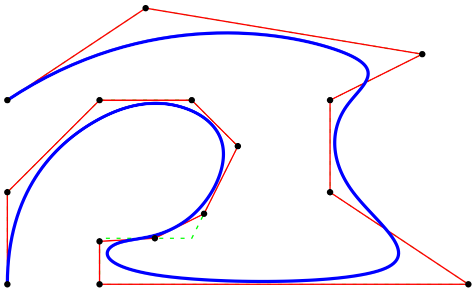
$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix} \right\}$$



Knot Insertion: Sample

- Clamped uniform B-spline of degree three for 14 control points and knot vector with 18 knots: $\tau := (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11, 11)$.

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix} \right\}$$



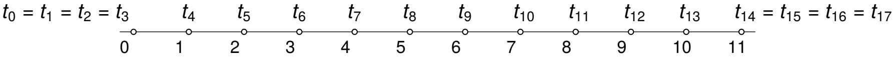
- New 15 control points for 19 knots $\tau^\star := (0, 0, 0, 0, 1, 2, 3, 4, 4.2, 5, 6, 7, 8, 9, 10, 11, 11, 11, 11)$:

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \begin{pmatrix} 5.2667 \\ 2.5333 \end{pmatrix}, \begin{pmatrix} 4.2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1.9333 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 11 \\ 1 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 5 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix} \right\}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Insertion: Sample

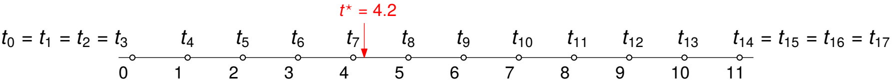
- Old 18 knots:



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Insertion: Sample

- Old 18 knots:

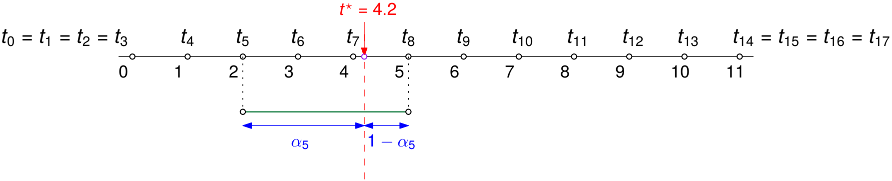


- For $t^\star := 4.2 = 4\frac{1}{5}$ we have $j = 7$ and $j - k + 1 = 5$,

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Insertion: Sample

- Old 18 knots:

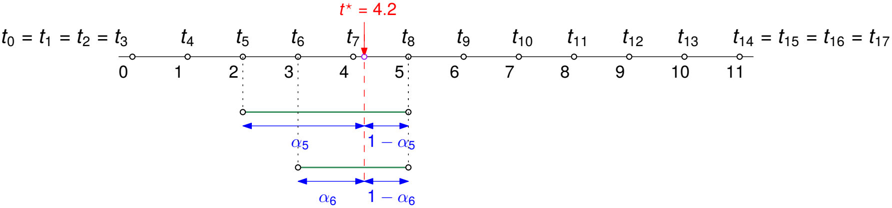


- For $t^\star := 4.2 = 4\frac{1}{5}$ we have $j = 7$ and $j - k + 1 = 5$, and get:

$$\alpha_5 := \frac{t^\star - t_5}{t_8 - t_5} = \frac{2\frac{1}{5}}{3} = \frac{11}{15} \qquad p_5^\star = (1 - \alpha_5)p_4 + \alpha_5 p_5 = \frac{1}{15}\begin{pmatrix} 79 \\ 38 \end{pmatrix} \approx \begin{pmatrix} 5.2667 \\ 2.5333 \end{pmatrix}$$

Knot Insertion: Sample

- Old 18 knots:

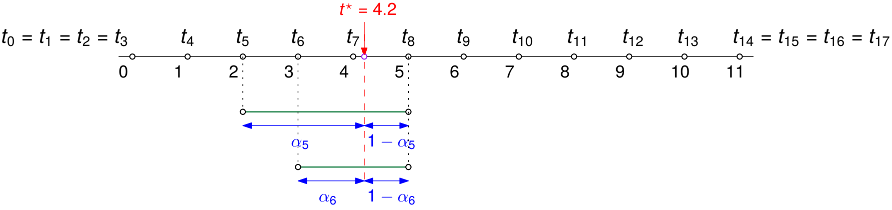


- For $t^\star := 4.2 = 4\frac{1}{5}$ we have $j = 7$ and $j - k + 1 = 5$, and get:

$$\alpha_5 := \frac{t^\star - t_5}{t_8 - t_5} = \frac{2\frac{1}{5}}{3} = \frac{11}{15} \qquad p_5^\star = (1-\alpha_5)p_4 + \alpha_5 p_5 = \frac{1}{15}\begin{pmatrix} 79 \\ 38 \end{pmatrix} \approx \begin{pmatrix} 5.2667 \\ 2.5333 \end{pmatrix}$$

$$\alpha_6 := \frac{t^\star - t_6}{t_9 - t_6} = \frac{1\frac{1}{5}}{3} = \frac{6}{15} \qquad p_6^\star = (1-\alpha_6)p_5 + \alpha_6 p_6 = \frac{1}{15}\begin{pmatrix} 63 \\ 30 \end{pmatrix} \approx \begin{pmatrix} 4.2 \\ 2 \end{pmatrix}$$

- Old 18 knots:



- For $t^\star := 4.2 = 4\frac{1}{5}$ we have $j = 7$ and $j - k + 1 = 5$, and get:

$$\alpha_5 := \frac{t^\star - t_5}{t_8 - t_5} = \frac{2\frac{1}{5}}{3} = \frac{11}{15} \qquad p_5^\star = (1-\alpha_5)p_4 + \alpha_5 p_5 = \frac{1}{15}\begin{pmatrix}79\\38\end{pmatrix} \approx \begin{pmatrix}5.2667\\2.5333\end{pmatrix}$$

$$\alpha_6 := \frac{t^\star - t_6}{t_9 - t_6} = \frac{1\frac{1}{5}}{3} = \frac{6}{15} \qquad p_6^\star = (1-\alpha_6)p_5 + \alpha_6 p_6 = \frac{1}{15}\begin{pmatrix}63\\30\end{pmatrix} \approx \begin{pmatrix}4.2\\2\end{pmatrix}$$

$$\alpha_7 := \frac{t^\star - t_7}{t_{10} - t_7} = \frac{\frac{1}{5}}{3} = \frac{1}{15} \qquad p_7^\star = (1-\alpha_7)p_6 + \alpha_7 p_7 = \frac{1}{15}\begin{pmatrix}45\\29\end{pmatrix} \approx \begin{pmatrix}3\\1.9333\end{pmatrix}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Knot Insertion and Deletion

- The so-called Oslo algorithm, developed by Cohen et al. [1980], is more general than Boehm's algorithm: It allows the insertion of several (possibly multiple) knots into a knot vector. (It is also substantially more complex, though.)
- An algorithm for the removal of a knot is due to Tiller [1992]. However, as pointed out by Tiller, knot removal and degree reduction result in an overspecified problem which, in general, can only be solved within some tolerance.

Lemma 159

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the uniform (unclamped) knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \le n$. If

$$p_0 = p_{n-k+1}, p_1 = p_{n-k+2}, \ldots, p_{k-2} = p_{n-1}, p_{k-1} = p_n$$

then $\mathcal{P}$ is C^{k-1} at the joining point $\mathcal{P}(t_k) = \lim_{t \nearrow t_{n+1}} \mathcal{P}(t)$.

Closed B-Spline Curve

Lemma 159

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the uniform (unclamped) knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. If

$$p_0 = p_{n-k+1}, p_1 = p_{n-k+2}, \ldots, p_{k-2} = p_{n-1}, p_{k-1} = p_n$$

then $\mathcal{P}$ is C^{k-1} at the joining point $\mathcal{P}(t_k) = \lim_{t \nearrow t_{n+1}} \mathcal{P}(t)$.

- Hence, wrapping around k control points achieves C^{k-1}-continuity at the joining point.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Closed B-Spline Curve

Lemma 159

Let $\mathcal{P}$ be a B-spline curve of degree k defined by $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ and the uniform (unclamped) knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$, for $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$. If

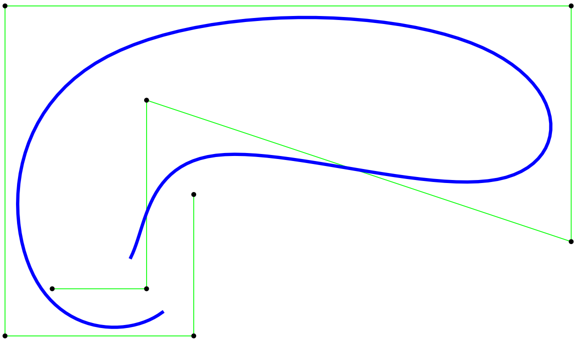
$$p_0 = p_{n-k+1}, p_1 = p_{n-k+2}, \ldots, p_{k-2} = p_{n-1}, p_{k-1} = p_n$$

then $\mathcal{P}$ is C^{k-1} at the joining point $\mathcal{P}(t_k) = \lim_{t \nearrow t_{n+1}} \mathcal{P}(t)$.

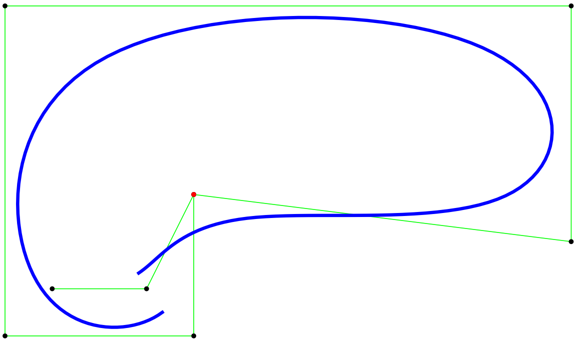
- Hence, wrapping around k control points achieves C^{k-1}-continuity at the joining point.
- A closed B-spline curve with C^{k-1}-continuity at the joining point can also be achieved by resorting to a periodic knot vector and wrapping around $k+2$ knots.

Closed B-Spline Curve

- Uniform B-spline of degree three for nine control points:
$$\left\{ \begin{pmatrix}4\\3\end{pmatrix}, \begin{pmatrix}4\\0\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0\\7\end{pmatrix}, \begin{pmatrix}12\\7\end{pmatrix}, \begin{pmatrix}12\\2\end{pmatrix}, \begin{pmatrix}3\\5\end{pmatrix}, \begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix} \right\}$$
Knot vector with 13 knots: $\tau := (0, \frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{5}{12}, \frac{6}{12}, \frac{7}{12}, \frac{8}{12}, \frac{9}{12}, \frac{10}{12}, \frac{11}{12}, 1)$.



- Uniform B-spline of degree three for nine control points:
  $$\left\{ \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 7 \end{pmatrix}, \begin{pmatrix} 12 \\ 7 \end{pmatrix}, \begin{pmatrix} 12 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$
  Knot vector with 13 knots: $\tau := (0, \frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{5}{12}, \frac{6}{12}, \frac{7}{12}, \frac{8}{12}, \frac{9}{12}, \frac{10}{12}, \frac{11}{12}, 1)$.

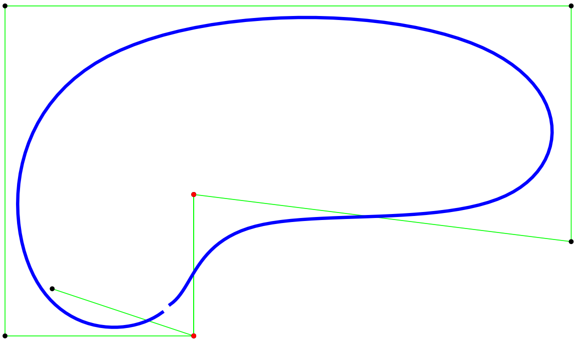


Closed B-Spline Curve

- Uniform B-spline of degree three for nine control points:

$$\left\{ \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 7 \end{pmatrix}, \begin{pmatrix} 12 \\ 7 \end{pmatrix}, \begin{pmatrix} 12 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$

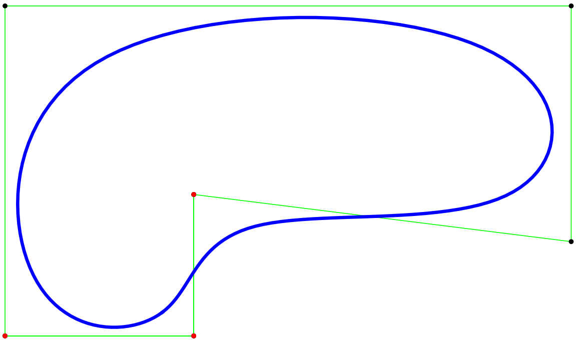
  Knot vector with 13 knots: $\tau := (0, \frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{5}{12}, \frac{6}{12}, \frac{7}{12}, \frac{8}{12}, \frac{9}{12}, \frac{10}{12}, \frac{11}{12}, 1)$.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Closed B-Spline Curve

- Uniform B-spline of degree three for nine control points:
  $\left\{ \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 7 \end{pmatrix}, \begin{pmatrix} 12 \\ 7 \end{pmatrix}, \begin{pmatrix} 12 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}$
  Knot vector with 13 knots: $\tau := (0, \frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{5}{12}, \frac{6}{12}, \frac{7}{12}, \frac{8}{12}, \frac{9}{12}, \frac{10}{12}, \frac{11}{12}, 1)$.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 160 (B-spline surface)

For $n, m \in \mathbb{N}$ and $k', k'' \in \mathbb{N}_0$ with $k' \leq n$ and $k'' \leq m$, consider a set of $(n+1) \times (m+1)$ control points with position vectors $p_{i,j} \in \mathbb{R}^3$ for $0 \leq i \leq n$ and $0 \leq j \leq m$, and let $\sigma := (s_0, s_1, \ldots, s_{n+k'+1})$ and $\tau := (t_0, t_1, \ldots, t_{m+k''+1})$ be two knot vectors. Then the *B-spline surface* relative to σ and τ with control net $(p_{i,j})_{i,j=0}^{n,m}$ is given by

$$\mathcal{S}(s,t) := \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,k',\sigma}(s) N_{j,k'',\tau}(t) p_{i,j} \qquad \text{for } s \in [s_{k'}, s_{n+1}[, \; t \in [t_{k''}, t_{m+1}[,$$

where $N_{i,k',\sigma}$ is the i-th B-spline basis function of degree k' relative to σ, and $N_{j,k'',\tau}$ is the j-th B-spline basis function of degree k'' relative to τ.

- Hence, a B-spline surface is another example of a tensor-product surface.

Properties of B-Spline Surfaces

Lemma 161 (Non-negativity)

With the setting of Def. 160, we have $N_{i,k',\sigma}(s)N_{j,k'',\tau}(t) \geq 0$ for all (permissible) $i, j \in \mathbb{Z}$ and $k', k'' \in \mathbb{N}_0$, and all real s, t.

Properties of B-Spline Surfaces

Lemma 161 (Non-negativity)

With the setting of Def. 160, we have $N_{i,k',\sigma}(s)N_{j,k'',\tau}(t) \geq 0$ for all (permissible) $i, j \in \mathbb{Z}$ and $k', k'' \in \mathbb{N}_0$, and all real s, t.

Lemma 162 (Partition of unity)

With the setting of Def. 160, we have

$$\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,k',\sigma}(s)N_{j,k'',\tau}(t) = 1$$

for all $s \in [s_{k'}, s_{n+1}[$, $t \in [t_{k''}, t_{m+1}[$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of B-Spline Surfaces

Lemma 161 (Non-negativity)

With the setting of Def. 160, we have $N_{i,k',\sigma}(s)N_{j,k'',\tau}(t) \geq 0$ for all (permissible) $i, j \in \mathbb{Z}$ and $k', k'' \in \mathbb{N}_0$, and all real s, t.

Lemma 162 (Partition of unity)

With the setting of Def. 160, we have

$$\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,k',\sigma}(s)N_{j,k'',\tau}(t) = 1$$

for all $s \in [s_{k'}, s_{n+1}[$, $t \in [t_{k''}, t_{m+1}[$.

Lemma 163 (Strong convex hull property)

With the setting of Def. 160, for $i, j \in \mathbb{N}$ with $k' \leq i \leq n$ and $k'' \leq j \leq m$ we have

$$\mathcal{S}|_{[s_i, s_{i+1}[\times[t_j, t_{j+1}[} \subset \text{CH}(\{p_{l',l''} : i - k' \leq l' \leq i \wedge j - k'' \leq l'' \leq j\}).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 164 (Local control)

With the setting of Def. 160, for $i, j \in \mathbb{N}$ with $k' \le i \le n$ and $k'' \le j \le m$ we have that

$$\mathcal{S}|_{[s_i, s_{i+1}[\times[t_j, t_{j+1}[} \quad \text{depends only on} \quad \{p_{l', l''} : i - k' \le l' \le i \wedge j - k'' \le l'' \le j\}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 164 (Local control)

With the setting of Def. 160, for $i, j \in \mathbb{N}$ with $k' \le i \le n$ and $k'' \le j \le m$ we have that

$$\mathcal{S}|_{[s_i, s_{i+1}[\times[t_j, t_{j+1}[} \quad \text{depends only on} \quad \{p_{l',l''} : i - k' \le l' \le i \wedge j - k'' \le l'' \le j\}.$$

Lemma 165 (Local modification scheme)

With the setting of Def. 160, a modification of the position of $p_{i,j}$ changes $\mathcal{S}$ only in the parameter domain $[s_i, s_{i+k'+1}[\times[t_j, t_{j+k''+1}[$, for $i \in \{0, 1, \ldots, n\}$ and $j \in \{0, 1, \ldots, m\}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of B-Spline Surfaces

Lemma 164 (Local control)

With the setting of Def. 160, for $i, j \in \mathbb{N}$ with $k' \leq i \leq n$ and $k'' \leq j \leq m$ we have that

$$\mathcal{S}|_{[s_i, s_{i+1}[\times[t_j, t_{j+1}[} \quad \text{depends only on} \quad \{p_{l',l''} : i - k' \leq l' \leq i \wedge j - k'' \leq l'' \leq j\}.$$

Lemma 165 (Local modification scheme)

With the setting of Def. 160, a modification of the position of $p_{i,j}$ changes $\mathcal{S}$ only in the parameter domain $[s_i, s_{i+k'+1}[\times[t_j, t_{j+k''+1}[$, for $i \in \{0, 1, \ldots, n\}$ and $j \in \{0, 1, \ldots, m\}$.

Lemma 166 (Affine invariance)

Any B-spline representation is affinely invariant, i.e., given any affine map π, the image surface $\pi(\mathcal{S})$ of a B-spline surface $\mathcal{S}$ with control points $p_{i,j}$ has the control points $\pi(p_{i,j})$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Clamping of a B-Spline Surface

- A B-spline surface can be clamped by repeating the same knot values in one direction of the parameters (i.e., in s or t).

Clamping of a B-Spline Surface

- A B-spline surface can be clamped by repeating the same knot values in one direction of the parameters (i.e., in s or t).
- We can also close the surface by recycling the control points.

Clamping of a B-Spline Surface

- A B-spline surface can be clamped by repeating the same knot values in one direction of the parameters (i.e., in s or t).
- We can also close the surface by recycling the control points.
- If a B-spline surface is closed in one direction, then the surface becomes a tube.
- If a B-spline surface is closed in two directions, then the surface becomes a torus.

Clamping of a B-Spline Surface

- A B-spline surface can be clamped by repeating the same knot values in one direction of the parameters (i.e., in s or t).
- We can also close the surface by recycling the control points.
- If a B-spline surface is closed in one direction, then the surface becomes a tube.
- If a B-spline surface is closed in two directions, then the surface becomes a torus.
- Other topologies are more difficult to handle, such as a ball or a double torus.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a B-Spline Surface

- Five easy steps to calculate a point on a B-spline patch for (s, t)
  1. Find the knot span in which s lies, i.e., find i such that $s \in [s_i, s_{i+1}[$.
  2. Evaluate the non-zero basis functions $N_{i-k',k'}(s), \ldots, N_{i,k'}(s)$.
  3. Find the knot span in which t lies, i.e., find j such that $t \in [t_j, t_{j+1}[$.
  4. Evaluate the non-zero basis functions $N_{j-k'',k''}(t), \ldots, N_{j,k''}(t)$.
  5. Multiply $N_{i',k'}(s)$ with $N_{j',k''}(t)$ and with the control point $p_{i',j'}$, for $i' \in \{i-k', \ldots, i\}$ and $j' \in \{j-k'', \ldots, j\}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Evaluation of a B-Spline Surface

- Five easy steps to calculate a point on a B-spline patch for (s, t)
  1. Find the knot span in which s lies, i.e., find i such that $s \in [s_i, s_{i+1}[$.
  2. Evaluate the non-zero basis functions $N_{i-k',k'}(s), \ldots, N_{i,k'}(s)$.
  3. Find the knot span in which t lies, i.e., find j such that $t \in [t_j, t_{j+1}[$.
  4. Evaluate the non-zero basis functions $N_{j-k'',k''}(t), \ldots, N_{j,k''}(t)$.
  5. Multiply $N_{i',k'}(s)$ with $N_{j',k''}(t)$ and with the control point $p_{i',j'}$, for $i' \in \{i - k', \ldots, i\}$ and $j' \in \{j - k'', \ldots, j\}$.
- Alternatively, one can apply an appropriate generalization of de Boor's algorithm.

4 B-Spline Curves and Surfaces

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

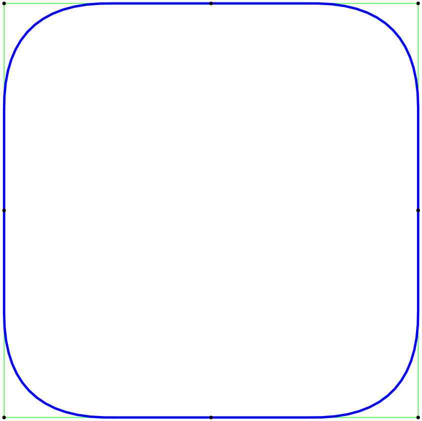
Motivation

- Can we use a B-spline curve to represent a circular arc?

Motivation

- Can we use a B-spline curve to represent a circular arc?

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

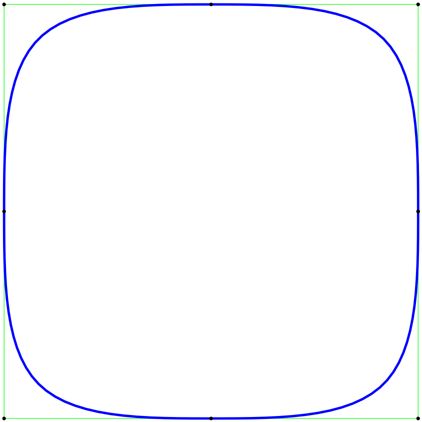


- uniform knots, degree 2

Motivation

- Can we use a B-spline curve to represent a circular arc?

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

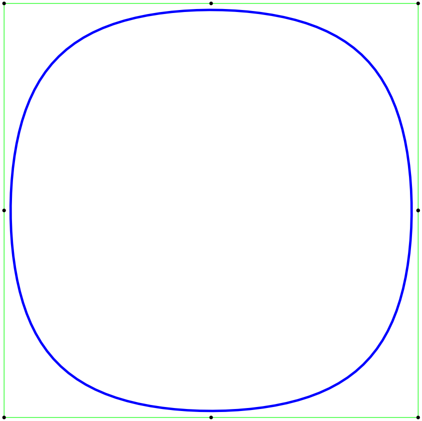


- uniform knots, degree 3

Motivation

- Can we use a B-spline curve to represent a circular arc?

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

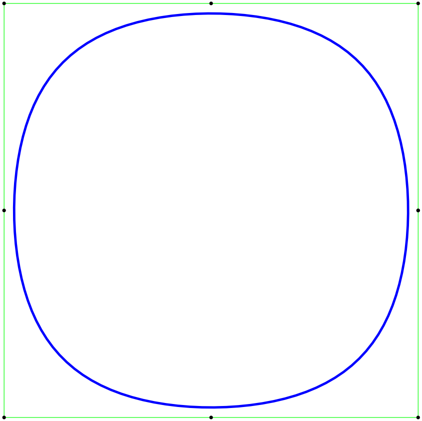


- uniform knots, degree 6

Motivation

- Can we use a B-spline curve to represent a circular arc?

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$



- uniform knots, degree 8
- close to a circle, but still no circle!

Non-Uniform Rational B-Splines

Definition 167 (NURBS curve)

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then a *rational B-spline curve* of *degree* k (and *order* $k+1$) relative to τ with control points $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{N}(t) := \frac{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i p_i}{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i} \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where $N_{i,k,\tau}$ is the i-th B-spline basis function of degree k relative to τ, and for some *weights* $w_i \in \mathbb{R}^+$, for all $i \in \{0, 1, \ldots, n\}$.

Definition 167 (NURBS curve)

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then a *rational B-spline curve* of *degree* k (and *order* $k+1$) relative to τ with control points $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{N}(t) := \frac{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i p_i}{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i} \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where $N_{i,k,\tau}$ is the i-th B-spline basis function of degree k relative to τ, and for some *weights* $w_i \in \mathbb{R}^+$, for all $i \in \{0, 1, \ldots, n\}$.

- If all $w_i := 1$ then we obtain the standard B-spline curve. (Recall the Partition of Unity, Cor. 137.)

Definition 167 (NURBS curve)

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then a *rational B-spline curve* of *degree* k (and *order* $k+1$) relative to τ with control points $p_0, p_1, \ldots, p_n$ is given by

$$\mathcal{N}(t) := \frac{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i p_i}{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i} \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where $N_{i,k,\tau}$ is the i-th B-spline basis function of degree k relative to τ, and for some *weights* $w_i \in \mathbb{R}^+$, for all $i \in \{0, 1, \ldots, n\}$.

- If all $w_i := 1$ then we obtain the standard B-spline curve. (Recall the Partition of Unity, Cor. 137.)
- Both the numerator and the denominator are (piecewise) polynomials of degree k. Hence, $\mathcal{N}$ is a piecewise rational curve of degree k.

Definition 167 (NURBS curve)

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then a *rational B-spline curve* of *degree* k (and *order* $k+1$) relative to τ with control points $p_0, p_1, \ldots, p_n$ is given by

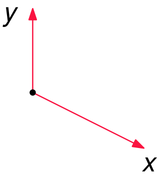
$$\mathcal{N}(t) := \frac{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i p_i}{\sum_{i=0}^{n} N_{i,k,\tau}(t) w_i} \qquad \text{for } t \in [t_k, t_{n+1}[,$$

where $N_{i,k,\tau}$ is the i-th B-spline basis function of degree k relative to τ, and for some *weights* $w_i \in \mathbb{R}^+$, for all $i \in \{0, 1, \ldots, n\}$.

- If all $w_i := 1$ then we obtain the standard B-spline curve. (Recall the Partition of Unity, Cor. 137.)
- Both the numerator and the denominator are (piecewise) polynomials of degree k. Hence, $\mathcal{N}$ is a piecewise rational curve of degree k.
- In general, the weights w_i are required to be positive; a zero weight effectively turns off a control point, and can be used for so-called infinite control points [Piegl 1987].

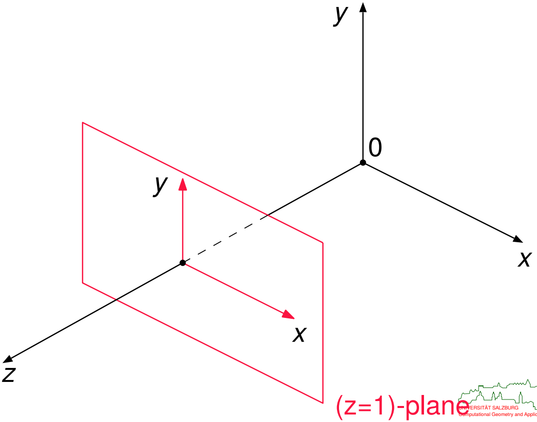
Geometric Interpretation of NURBS: Homogeneous Coordinates

- $\mathbb{R}^2$



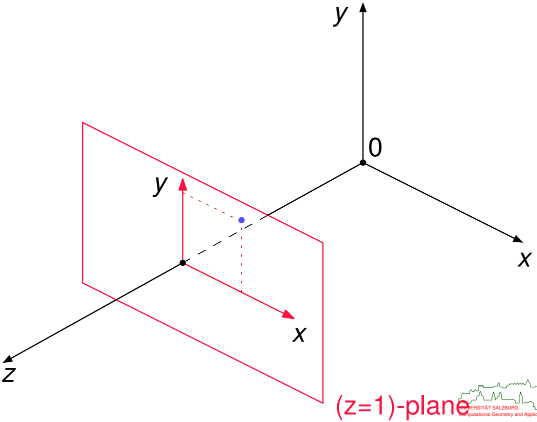
Geometric Interpretation of NURBS: Homogeneous Coordinates

- $\mathbb{R}^2$ is embedded into $\mathbb{R}^3$ by identifying it with the plane $z = 1$.



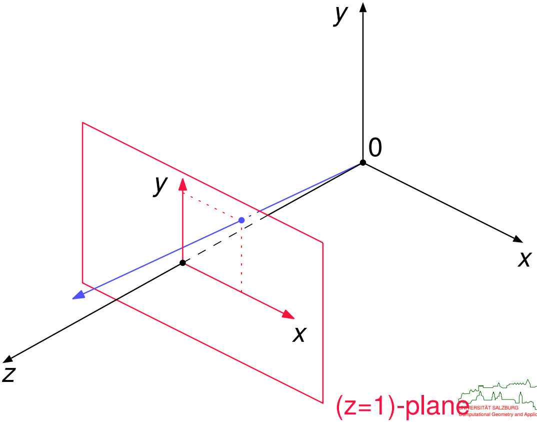
Geometric Interpretation of NURBS: Homogeneous Coordinates

- $\mathbb{R}^2$ is embedded into $\mathbb{R}^3$ by identifying it with the plane $z = 1$.
- We identify the point $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ with $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in \mathbb{R}^3$



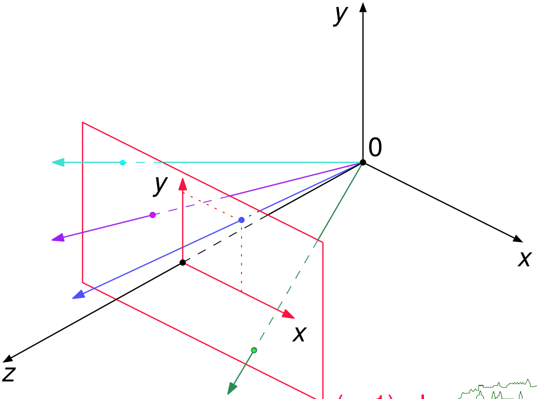
Geometric Interpretation of NURBS: Homogeneous Coordinates

- $\mathbb{R}^2$ is embedded into $\mathbb{R}^3$ by identifying it with the plane $z = 1$.
- We identify the point $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ with $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in \mathbb{R}^3$ or with $\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \end{pmatrix} \in \mathbb{R}^3$ for $w \neq 0$.

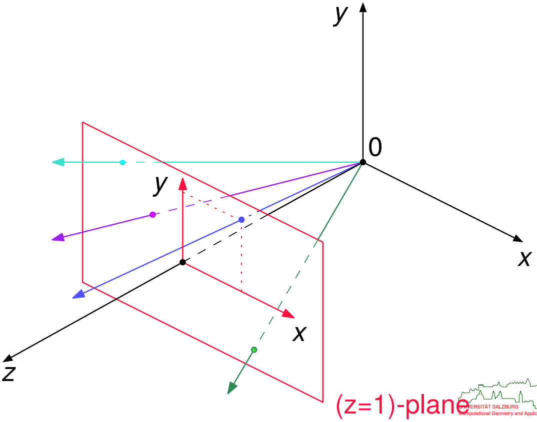


Geometric Interpretation of NURBS: Homogeneous Coordinates

- $\mathbb{R}^2$ is embedded into $\mathbb{R}^3$ by identifying it with the plane $z = 1$.
- We identify the point $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ with $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in \mathbb{R}^3$ or with $\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \end{pmatrix} \in \mathbb{R}^3$ for $w \neq 0$.
- Same for other points.



- $\mathbb{R}^2$ is embedded into $\mathbb{R}^3$ by identifying it with the plane $z = 1$.
- We identify the point $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ with $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in \mathbb{R}^3$ or with $\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \end{pmatrix} \in \mathbb{R}^3$ for $w \neq 0$.
- Same for other points.
- All points on a particular line through the origin in $\mathbb{R}^3$ represent the same point in $\mathbb{R}^2$.



Geometric Interpretation of NURBS: Homogeneous Coordinates

Definition 168 (Homogeneous coordinates, Dt.: homogene Koordinaten)

Homogeneous coordinates of $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ are given by $\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \end{pmatrix} \in \mathbb{R}^3$, for $w \neq 0$,

while

the *inhomogeneous coordinates* of $\begin{pmatrix} x \\ y \\ w \end{pmatrix} \in \mathbb{R}^3$ are given by $\begin{pmatrix} x/w \\ y/w \end{pmatrix} \in \mathbb{R}^2$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Geometric Interpretation of NURBS: Homogeneous Coordinates

Definition 168 (Homogeneous coordinates, Dt.: homogene Koordinaten)

Homogeneous coordinates of $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ are given by $\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \end{pmatrix} \in \mathbb{R}^3$, for $w \neq 0$,

while

the *inhomogeneous coordinates* of $\begin{pmatrix} x \\ y \\ w \end{pmatrix} \in \mathbb{R}^3$ are given by $\begin{pmatrix} x/w \\ y/w \end{pmatrix} \in \mathbb{R}^2$.

- For $p_i := \begin{pmatrix} x_i \\ y_i \end{pmatrix} \in \mathbb{R}^2$, let $p_i^w := \begin{pmatrix} w_i x_i \\ w_i y_i \\ w_i \end{pmatrix} \in \mathbb{R}^3$, for all $i \in \{0, 1, \ldots, n\}$.

Definition 168 (Homogeneous coordinates, Dt.: homogene Koordinaten)

Homogeneous coordinates of $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ are given by $\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \end{pmatrix} \in \mathbb{R}^3$, for $w \neq 0$,

while

the *inhomogeneous coordinates* of $\begin{pmatrix} x \\ y \\ w \end{pmatrix} \in \mathbb{R}^3$ are given by $\begin{pmatrix} x/w \\ y/w \end{pmatrix} \in \mathbb{R}^2$.

- For $p_i := \begin{pmatrix} x_i \\ y_i \end{pmatrix} \in \mathbb{R}^2$, let $p_i^w := \begin{pmatrix} w_i x_i \\ w_i y_i \\ w_i \end{pmatrix} \in \mathbb{R}^3$, for all $i \in \{0, 1, \ldots, n\}$.
- Now consider

$$\mathcal{N}^w(t) := \sum_{i=0}^{n} N_{i,k}(t)\, p_i^w = \begin{pmatrix} \sum_{i=0}^{n} N_{i,k}(t)(w_i x_i) \\ \sum_{i=0}^{n} N_{i,k}(t)(w_i y_i) \\ \sum_{i=0}^{n} N_{i,k}(t) w_i \end{pmatrix}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 168 (Homogeneous coordinates, Dt.: homogene Koordinaten)

Homogeneous coordinates of $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ are given by $\begin{pmatrix} w \cdot x \\ w \cdot y \\ w \end{pmatrix} \in \mathbb{R}^3$, for $w \neq 0$,

while

the *inhomogeneous coordinates* of $\begin{pmatrix} x \\ y \\ w \end{pmatrix} \in \mathbb{R}^3$ are given by $\begin{pmatrix} x/w \\ y/w \end{pmatrix} \in \mathbb{R}^2$.

- For $p_i := \begin{pmatrix} x_i \\ y_i \end{pmatrix} \in \mathbb{R}^2$, let $p_i^w := \begin{pmatrix} w_i x_i \\ w_i y_i \\ w_i \end{pmatrix} \in \mathbb{R}^3$, for all $i \in \{0, 1, \ldots, n\}$.
- Now consider

$$\mathcal{N}^w(t) := \sum_{i=0}^{n} N_{i,k}(t)\, p_i^w = \begin{pmatrix} \sum_{i=0}^{n} N_{i,k}(t)(w_i x_i) \\ \sum_{i=0}^{n} N_{i,k}(t)(w_i y_i) \\ \sum_{i=0}^{n} N_{i,k}(t) w_i \end{pmatrix}.$$

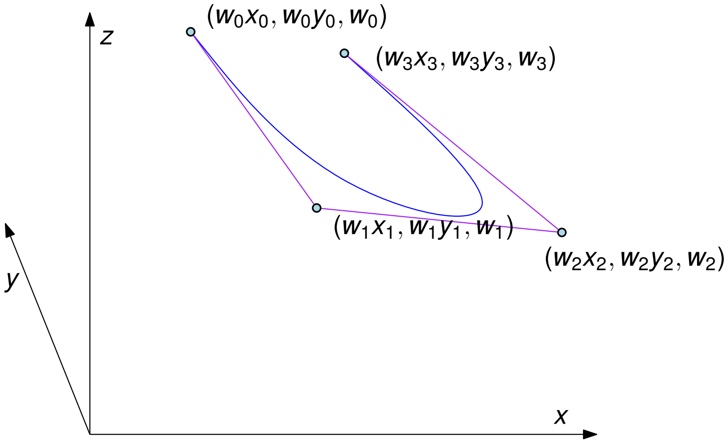
- Dividing the first two coordinates of $\mathcal{N}^w$ by its third coordinate equals the (central) projection of $\mathcal{N}^w$ to the plane $z = 1$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Geometric Interpretation of NURBS

Projection onto $z = 1$

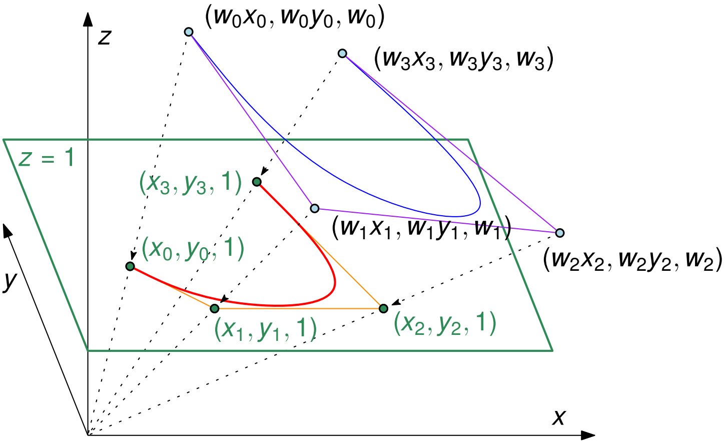
A NURBS curve in $\mathbb{R}^d$ is the projection of a B-spline curve in $\mathbb{R}^{d+1}$.



Geometric Interpretation of NURBS

Projection onto $z = 1$

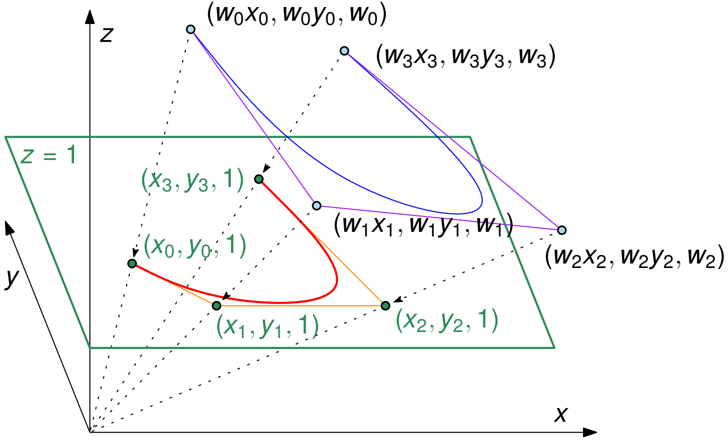
A NURBS curve in $\mathbb{R}^d$ is the projection of a B-spline curve in $\mathbb{R}^{d+1}$.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Projection onto $z = 1$

A NURBS curve in $\mathbb{R}^d$ is the projection of a B-spline curve in $\mathbb{R}^{d+1}$.



- Hence, NURBS curves inherit properties of B-spline curves.

Geometric Interpretation of NURBS

- Rational (inhomogeneous) parametrization of the unit circle in the plane:

$$x(t) := \frac{1 - t^2}{1 + t^2} \qquad \text{with } t \in \mathbb{R}.$$
$$y(t) := \frac{2t}{1 + t^2}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Geometric Interpretation of NURBS

- Rational (inhomogeneous) parametrization of the unit circle in the plane:

$$x(t) := \frac{1 - t^2}{1 + t^2} \qquad y(t) := \frac{2t}{1 + t^2} \qquad \text{with } t \in \mathbb{R}.$$

- Parametrization of the unit circle in the plane in homogeneous coordinates:

$$u(t) := 1 - t^2$$
$$v(t) := 2t$$
$$w(t) := 1 + t^2$$

NURBS Basis Functions

Definition 169 (NURBS basis function)

For $k \in \mathbb{N}_0$, weights $w_j > 0$ for all $j \in \{0, 1, \ldots, n\}$ and all (permissible) i, we define the i-th *NURBS basis function* of degree k as

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

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 169 (NURBS basis function)

For $k \in \mathbb{N}_0$, weights $w_j > 0$ for all $j \in \{0, 1, \ldots, n\}$ and all (permissible) i, we define the i-th *NURBS basis function* of degree k as

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

- We can re-write the equation (in Def. 167) for $\mathcal{N}(t)$ as

$$\mathcal{N}(t) = \sum_{i=0}^{n} R_{i,k}(t)p_i \qquad \text{for } t \in [t_k, t_{n+1}[.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

NURBS Basis Functions

Definition 169 (NURBS basis function)

For $k \in \mathbb{N}_0$, weights $w_j > 0$ for all $j \in \{0, 1, \ldots, n\}$ and all (permissible) i, we define the i-th *NURBS basis function* of degree k as

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

- We can re-write the equation (in Def. 167) for $\mathcal{N}(t)$ as

$$\mathcal{N}(t) = \sum_{i=0}^{n} R_{i,k}(t) p_i \qquad \text{for } t \in [t_k, t_{n+1}[.$$

- Since NURBS basis functions in $\mathbb{R}^d$ are given by the projection of B-spline basis functions in $\mathbb{R}^{d+1}$, we may expect that the properties of B-spline basis functions carry over to NURBS basis functions.

Lemma 170

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, let $\tau := (t_0, t_1, t_2, \ldots, t_{n+k+1})$ be a knot vector.

Lemma 170

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, let $\tau := (t_0, t_1, t_2, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold for all (permissible) values of $i \in \mathbb{N}_0$:

Non-negativity:

$$R_{i,k}(t) \geq 0 \qquad \text{for all real } t.$$

Lemma 170

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, let $\tau := (t_0, t_1, t_2, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold for all (permissible) values of $i \in \mathbb{N}_0$:

Non-negativity:

$$R_{i,k}(t) \geq 0 \qquad \text{for all real } t.$$

Local support:

$$R_{i,k}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Lemma 170

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \le n$, let $\tau := (t_0, t_1, t_2, \dots, t_{n+k+1})$ be a knot vector. Then the following properties hold for all (permissible) values of $i \in \mathbb{N}_0$:

Non-negativity:

$$R_{i,k}(t) \ge 0 \qquad \text{for all real } t.$$

Local support:

$$R_{i,k}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Local influence:

$$R_{j,k} \text{ non-zero over } [t_i, t_{i+1}[\quad \Rightarrow \quad j \in \{i-k, i-k+1, \dots, i\}.$$

Properties of NURBS Basis Functions

Lemma 170

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, let $\tau := (t_0, t_1, t_2, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold for all (permissible) values of $i \in \mathbb{N}_0$:

Non-negativity:

$$R_{i,k}(t) \geq 0 \qquad \text{for all real } t.$$

Local support:

$$R_{i,k}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Local influence:

$$R_{j,k} \text{ non-zero over } [t_i, t_{i+1}[\quad \Rightarrow \quad j \in \{i-k, i-k+1, \ldots, i\}.$$

Partition of unity:

$$\sum_{j=0}^{n} R_{j,k}(t) = 1 \qquad \text{for all } t \in [t_k, t_{n+1}[.$$

Lemma 170

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, let $\tau := (t_0, t_1, t_2, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold for all (permissible) values of $i \in \mathbb{N}_0$:

Non-negativity:

$$R_{i,k}(t) \geq 0 \qquad \text{for all real } t.$$

Local support:

$$R_{i,k}(t) = 0 \qquad \text{if } t \notin [t_i, t_{i+k+1}[.$$

Local influence:

$$R_{j,k} \text{ non-zero over } [t_i, t_{i+1}[\quad \Rightarrow \quad j \in \{i-k, i-k+1, \ldots, i\}.$$

Partition of unity:

$$\sum_{j=0}^{n} R_{j,k}(t) = 1 \qquad \text{for all } t \in [t_k, t_{n+1}[.$$

Continuity:

All NURBS basis functions of degree k are $k-r$ times continuously differentiable at a knot of multiplicity r, and $k-1$ times continuously differentiable everywhere else.

Lemma 171

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 171

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold:

Clamped interpolation: If τ is clamped then the NURBS curve $\mathcal{N}$ starts in p_0 and ends in p_n.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 171

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold:

Clamped interpolation: If τ is clamped then the NURBS curve $\mathcal{N}$ starts in p_0 and ends in p_n.

Variation diminishing property: If a straight line intersects the control polygon of $\mathcal{N}$ m times then it intersects $\mathcal{N}$ at most m times.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 171

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold:

Clamped interpolation: If τ is clamped then the NURBS curve $\mathcal{N}$ starts in p_0 and ends in p_n.

Variation diminishing property: If a straight line intersects the control polygon of $\mathcal{N}$ m times then it intersects $\mathcal{N}$ at most m times.

Strong convex hull property: For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{N}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

Lemma 171

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold:

Clamped interpolation: If τ is clamped then the NURBS curve $\mathcal{N}$ starts in p_0 and ends in p_n.

Variation diminishing property: If a straight line intersects the control polygon of $\mathcal{N}$ m times then it intersects $\mathcal{N}$ at most m times.

Strong convex hull property: For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{N}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

Local control: The NURBS curve $\mathcal{N}$ restricted to $[t_i, t_{i+1}[$ depends only on the positions of $p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 171

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector. Then the following properties hold:

Clamped interpolation: If τ is clamped then the NURBS curve $\mathcal{N}$ starts in p_0 and ends in p_n.

Variation diminishing property: If a straight line intersects the control polygon of $\mathcal{N}$ m times then it intersects $\mathcal{N}$ at most m times.

Strong convex hull property: For $i \in \mathbb{N}$ with $k \leq i \leq n$, we have

$$\mathcal{N}|_{[t_i, t_{i+1}[} \subset \mathrm{CH}(\{p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i\}).$$

Local control: The NURBS curve $\mathcal{N}$ restricted to $[t_i, t_{i+1}[$ depends only on the positions of $p_{i-k}, p_{i-k+1}, \ldots, p_{i-1}, p_i$.

Local modification scheme: A modification of the position of p_i changes $\mathcal{N}$ only in the parameter interval $[t_i, t_{i+k+1}[$, for $i \in \{0, 1, \ldots, n\}$.

Properties of NURBS Curves

Lemma 172 (Projective invariance)

Any NURBS curve is projectively invariant, i.e., given any projective transformation π, the image curve $\pi(\mathcal{N})$ of a NURBS curve $\mathcal{N}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of NURBS Curves

Lemma 172 (Projective invariance)

Any NURBS curve is projectively invariant, i.e., given any projective transformation π, the image curve $\pi(\mathcal{N})$ of a NURBS curve $\mathcal{N}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

Lemma 173

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector and $w_0, w_1, \ldots, w_n$ be weights.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of NURBS Curves

Lemma 172 (Projective invariance)

Any NURBS curve is projectively invariant, i.e., given any projective transformation π, the image curve $\pi(\mathcal{N})$ of a NURBS curve $\mathcal{N}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

Lemma 173

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector and $w_0, w_1, \ldots, w_n$ be weights. Then the following properties hold for all $i \in \{0, 1, \ldots, n\}$:

1. The weight w_i effects only the knot span $[t_i, t_{i+k+1}[$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 172 (Projective invariance)

Any NURBS curve is projectively invariant, i.e., given any projective transformation π, the image curve $\pi(\mathcal{N})$ of a NURBS curve $\mathcal{N}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

Lemma 173

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n + 1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector and $w_0, w_1, \ldots, w_n$ be weights. Then the following properties hold for all $i \in \{0, 1, \ldots, n\}$:

1. The weight w_i effects only the knot span $[t_i, t_{i+k+1}[$.
2. If w_i decreases (relative to the other weights) then the NURBS curve is pushed away from p_i.

Lemma 172 (Projective invariance)

Any NURBS curve is projectively invariant, i.e., given any projective transformation π, the image curve $\pi(\mathcal{N})$ of a NURBS curve $\mathcal{N}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

Lemma 173

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector and $w_0, w_1, \ldots, w_n$ be weights. Then the following properties hold for all $i \in \{0, 1, \ldots, n\}$:

1. The weight w_i effects only the knot span $[t_i, t_{i+k+1}[$.
2. If w_i decreases (relative to the other weights) then the NURBS curve is pushed away from p_i.
3. If $w_i = 0$ then p_i does not contribute to the NURBS curve.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Properties of NURBS Curves

Lemma 172 (Projective invariance)

Any NURBS curve is projectively invariant, i.e., given any projective transformation π, the image curve $\pi(\mathcal{N})$ of a NURBS curve $\mathcal{N}$ with control points $p_0, p_1, \ldots, p_n$ has the control points $\pi(p_0), \pi(p_1), \ldots, \pi(p_n)$.

Lemma 173

For $n \in \mathbb{N}$ and $k \in \mathbb{N}_0$ with $k \leq n$, consider a set of $n+1$ control points with position vectors $p_0, p_1, \ldots, p_n$ in the plane, and let $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ be a knot vector and $w_0, w_1, \ldots, w_n$ be weights. Then the following properties hold for all $i \in \{0, 1, \ldots, n\}$:

1. The weight w_i effects only the knot span $[t_i, t_{i+k+1}[$.
2. If w_i decreases (relative to the other weights) then the NURBS curve is pushed away from p_i.
3. If $w_i = 0$ then p_i does not contribute to the NURBS curve.
4. If w_i increases (relative to the other weights) then the NURBS curve is pulled towards p_i.

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

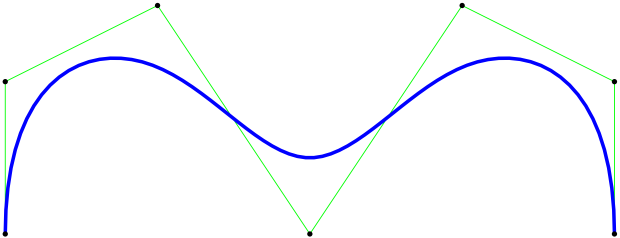
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

  Weights:

$$(1, 1, 1, 1, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

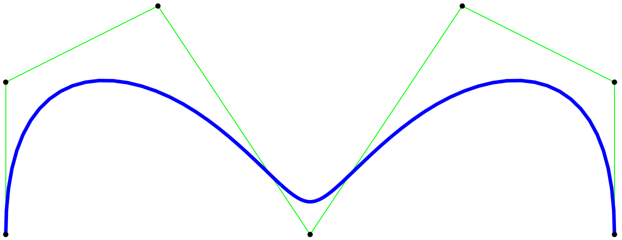
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

Weights:

$$(1, 1, 1, 3, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

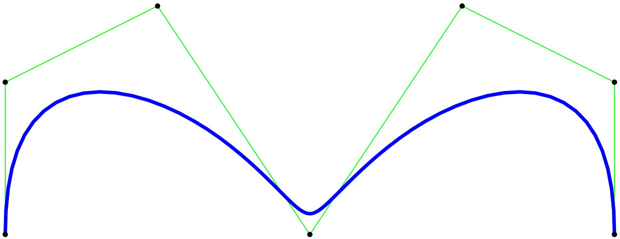
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

  Weights:

$$(1, 1, 1, 5, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

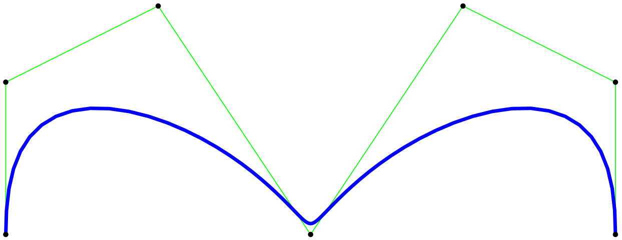
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

Weights:

$$(1, 1, 1, 10, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

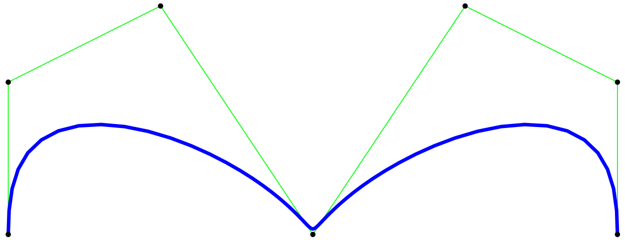
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

Weights:

$$(1, 1, 1, 20, 1, 1, 1)$$



- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

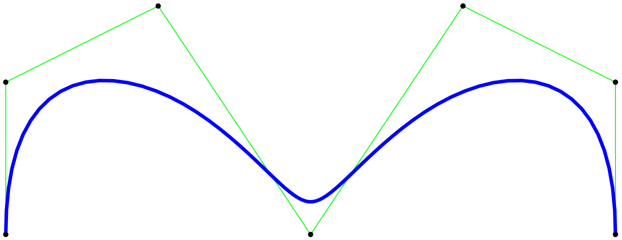
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

Weights:

$$(1, 1, 1, 1, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

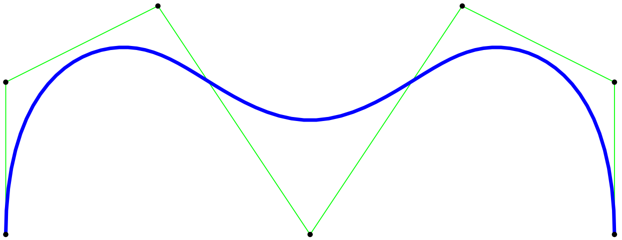
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

Weights:

$$(1, 1, 1, 0.5, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

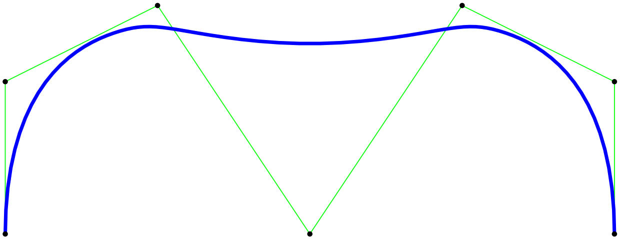
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

Weights:

$$(1, 1, 1, 0.1, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

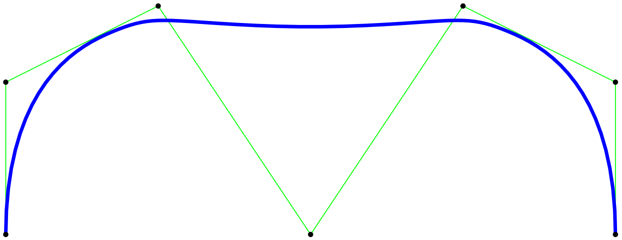
$$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector:

$$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

  Weights:

$$(1, 1, 1, 0.05, 1, 1, 1)$$



Sample NURBS Curve

- Clamped uniform rational B-spline of degree three for a control polygon with seven vertices:

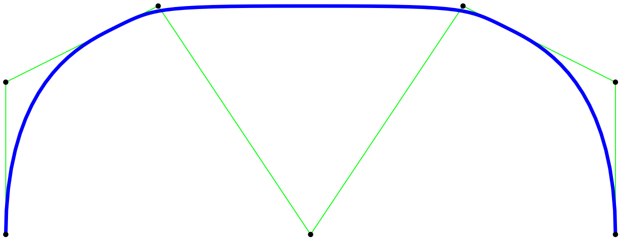
  $$\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 8 \\ 0 \end{pmatrix} \right\}$$

  Knot vector:

  $$\tau := (0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4)$$

  Weights:

  $$(1, 1, 1, 0, 1, 1, 1)$$



Conics Modeled by NURBS

- NURBS can represent all conic curves — circle, ellipse, parabola, hyperbola — exactly.

Conics Modeled by NURBS

- NURBS can represent all conic curves — circle, ellipse, parabola, hyperbola — exactly.
- Conics are quadratic curves.
- Hence, consider three control points p_0, p_1, p_2 and the following quadratic NURBS curve

$$\mathcal{N}_2(t) := \frac{\sum_{i=0}^{2} N_{i,2}(t)\; w_i\; p_i}{\sum_{i=0}^{2} N_{i,2}(t)\; w_i} \qquad \text{with } \tau := (0,0,0,1,1,1),$$

i.e., a rational Bézier curve of degree two over $[0, 1]$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Conics Modeled by NURBS

- NURBS can represent all conic curves — circle, ellipse, parabola, hyperbola — exactly.
- Conics are quadratic curves.
- Hence, consider three control points p_0, p_1, p_2 and the following quadratic NURBS curve

$$\mathcal{N}_2(t) := \frac{\sum_{i=0}^{2} N_{i,2}(t)\ w_i\ p_i}{\sum_{i=0}^{2} N_{i,2}(t)\ w_i} \qquad \text{with } \tau := (0, 0, 0, 1, 1, 1),$$

  i.e., a rational Bézier curve of degree two over $[0, 1]$.
- In expanded form we get

$$\mathcal{N}_2(t) = \frac{(1-t)^2 w_0 p_0 + 2t(1-t) w_1 p_1 + t^2 w_2 p_2}{(1-t)^2 w_0 + 2t(1-t) w_1 + t^2 w_2}.$$

Conics Modeled by NURBS

- NURBS can represent all conic curves — circle, ellipse, parabola, hyperbola — exactly.
- Conics are quadratic curves.
- Hence, consider three control points p_0, p_1, p_2 and the following quadratic NURBS curve

$$\mathcal{N}_2(t) := \frac{\sum_{i=0}^{2} N_{i,2}(t)\; w_i\; p_i}{\sum_{i=0}^{2} N_{i,2}(t)\; w_i} \qquad \text{with } \tau := (0,0,0,1,1,1),$$

  i.e., a rational Bézier curve of degree two over $[0,1]$.
- In expanded form we get

$$\mathcal{N}_2(t) = \frac{(1-t)^2 w_0 p_0 + 2t(1-t) w_1 p_1 + t^2 w_2 p_2}{(1-t)^2 w_0 + 2t(1-t) w_1 + t^2 w_2}.$$

- Can we come up with conditions for w_0, w_1, w_2 that allow to characterize the type of curve represented by $\mathcal{N}_2$?

Conics Modeled by NURBS

Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

- Clamped uniform rational B-spline $\mathcal{N}_2$ of degree two with three control vertices

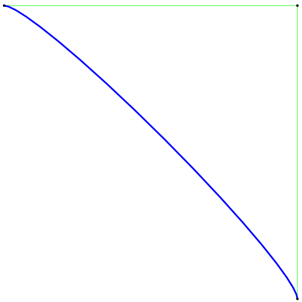
$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and knots

$$\tau := (0, 0, 0, 1, 1, 1)$$

and weights:

$(1, 1/10, 1)$, hence $\rho < 1$.



Conics Modeled by NURBS

Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

- Clamped uniform rational B-spline $\mathcal{N}_2$ of degree two with three control vertices

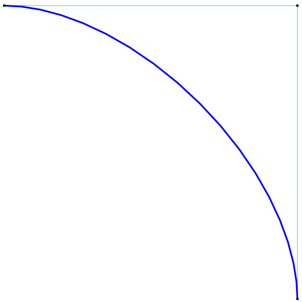
$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and knots

$$\tau := (0, 0, 0, 1, 1, 1)$$

and weights:

$$(1, 1/2, 1), \quad \text{hence } \rho < 1.$$



Conics Modeled by NURBS

Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

- Clamped uniform rational B-spline $\mathcal{N}_2$ of degree two with three control vertices

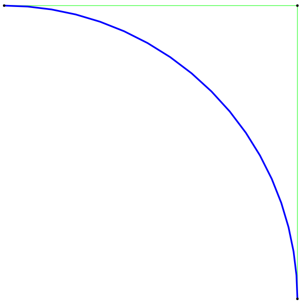
$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and knots

$$\tau := (0, 0, 0, 1, 1, 1)$$

and weights:

$$(1, 1/\sqrt{2}, 1), \quad \text{hence } \rho < 1.$$



Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

- Clamped uniform rational B-spline $\mathcal{N}_2$ of degree two with three control vertices

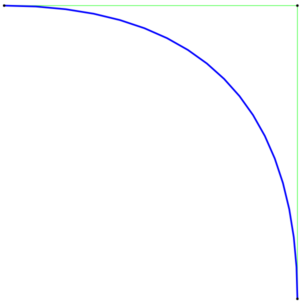
$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and knots

$$\tau := (0, 0, 0, 1, 1, 1)$$

and weights:

$(1, 1, 1)$, hence $\rho = 1$.



Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

- Clamped uniform rational B-spline $\mathcal{N}_2$ of degree two with three control vertices

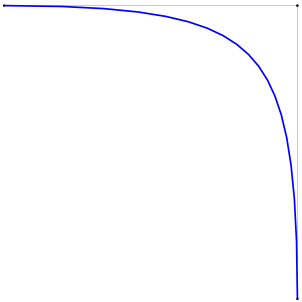
$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and knots

$$\tau := (0, 0, 0, 1, 1, 1)$$

and weights:

$(1, 2, 1)$, hence $\rho > 1$.



Conics Modeled by NURBS

Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

- Clamped uniform rational B-spline $\mathcal{N}_2$ of degree two with three control vertices

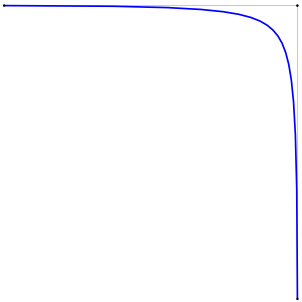
  $$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

  and knots

  $$\tau := (0, 0, 0, 1, 1, 1)$$

  and weights:

  $$(1, 5, 1), \quad \text{hence } \rho > 1.$$



Lemma 174

The *conic shape factor*, ρ, determines the type of conic represented by $\mathcal{N}_2$:

$$\rho := \frac{w_1^2}{w_0 w_2} \begin{cases} < 1 & \ldots \quad \mathcal{N}_2 \text{ is an elliptic curve,} \\ = 1 & \ldots \quad \mathcal{N}_2 \text{ is a parabolic curve,} \\ > 1 & \ldots \quad \mathcal{N}_2 \text{ is a hyperbolic curve.} \end{cases}$$

- Clamped uniform rational B-spline $\mathcal{N}_2$ of degree two with three control vertices

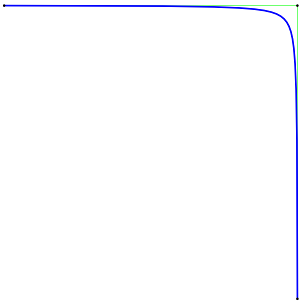
$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and knots

$$\tau := (0, 0, 0, 1, 1, 1)$$

and weights:

$(1, 10, 1)$, hence $\rho > 1$.



Lemma 175

The quadratic NURBS curve $\mathcal{N}_2$ represents a circular arc

- if the control points p_0, p_1, p_2 form an isosceles triangle, and
- if the weights are set as follows:

$$w_0 := 1 \qquad w_1 := \frac{\|p_0 - p_2\|}{2 \cdot \|p_0 - p_1\|} \qquad w_2 := 1$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Conics Modeled by NURBS

Lemma 175

The quadratic NURBS curve $\mathcal{N}_2$ represents a circular arc

- if the control points p_0, p_1, p_2 form an isosceles triangle, and
- if the weights are set as follows:

$$w_0 := 1 \qquad w_1 := \frac{\|p_0 - p_2\|}{2 \cdot \|p_0 - p_1\|} \qquad w_2 := 1$$

- The weight w_1 is related to the central angle φ subtended by the arc: $w_1 = \cos(\varphi/2)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Conics Modeled by NURBS

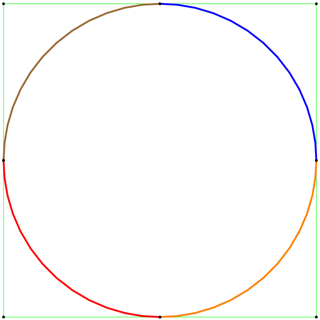
Lemma 175

The quadratic NURBS curve $\mathcal{N}_2$ represents a circular arc

- if the control points p_0, p_1, p_2 form an isosceles triangle, and
- if the weights are set as follows:

$$w_0 := 1 \qquad w_1 := \frac{\|p_0 - p_2\|}{2 \cdot \|p_0 - p_1\|} \qquad w_2 := 1$$

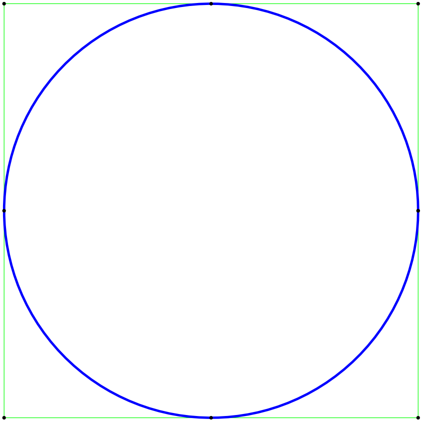
- The weight w_1 is related to the central angle φ subtended by the arc: $w_1 = \cos(\varphi/2)$.
- We can join four quarter-circle NURBS to form a full circle.
- In this case, the isosceles triangles defining the quarter circles need to add up to a square.



Conics Modeled by NURBS

- It is also possible to construct a circle by a single NURBS curve.

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$



Knots:

$$(0, 0, 0, \frac{\pi}{2}, \frac{\pi}{2}, \pi, \pi, \frac{3\pi}{2}, \frac{3\pi}{2}, 2\pi, 2\pi, 2\pi)$$

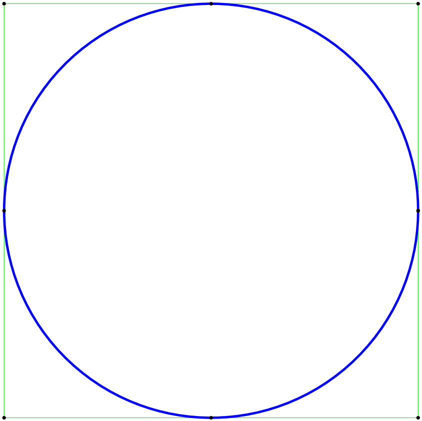
Weights:

$$(1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1)$$

Conics Modeled by NURBS

- It is also possible to construct a circle by a single NURBS curve.

$$\left\{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix}\right\}$$



Knots:

$$\left(0,0,0,\frac{\pi}{2},\frac{\pi}{2},\pi,\pi,\frac{3\pi}{2},\frac{3\pi}{2},2\pi,2\pi,2\pi\right)$$

Weights:

$$\left(1,\frac{1}{\sqrt{2}},1,\frac{1}{\sqrt{2}},1,\frac{1}{\sqrt{2}},1,\frac{1}{\sqrt{2}},1\right)$$

- Note: The positioning of the control points ensures that the first derivative is continuous, despite of double knots.
- Note: $\mathcal{N}(t) \neq (\cos t, \sin t)$ for $t \neq \frac{m\cdot\pi}{4}$.

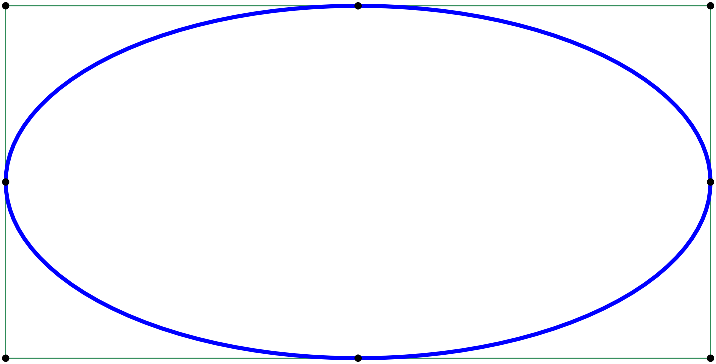
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Conics Modeled by NURBS

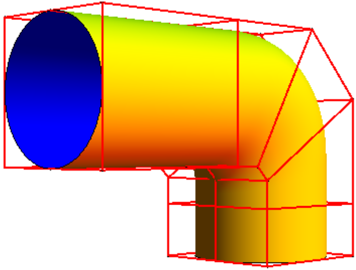
- Applying an affine transformation to the control points yields an ellipse.

$$\left\{\begin{pmatrix}2\\0\end{pmatrix},\begin{pmatrix}2\\1\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}-2\\1\end{pmatrix},\begin{pmatrix}-2\\0\end{pmatrix},\begin{pmatrix}-2\\-1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}2\\-1\end{pmatrix},\begin{pmatrix}2\\0\end{pmatrix}\right\}$$

Knots: $(0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4)$ Weights: $(1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1)$



Sample NURBS Surface

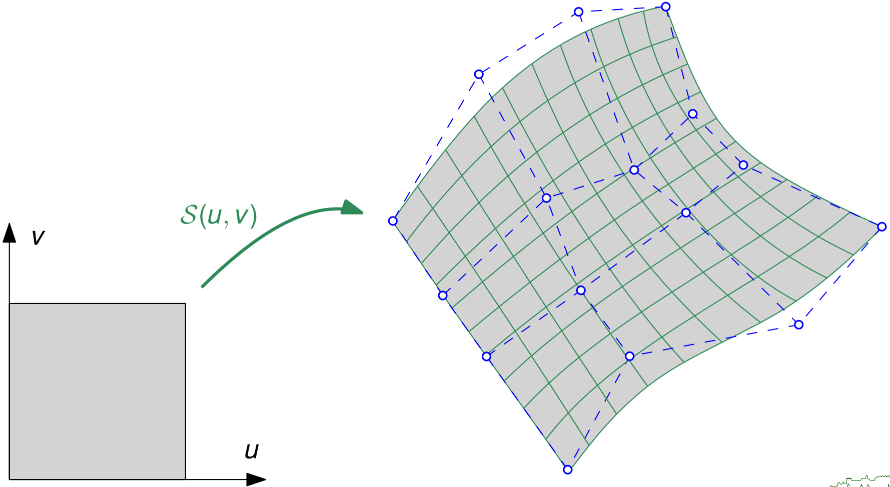


5 **Subdivision Methods**

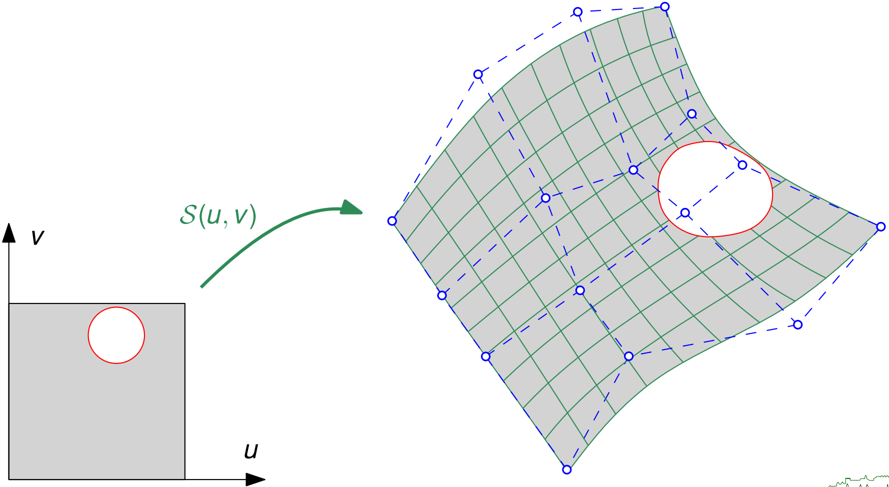
- Basics
- Subdivision Surfaces

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Consider a NURBS surface.



- Consider a NURBS surface. How could you intersect a cyclinder with it to cut out a spherical hole?



Problems of NURBS Surfaces: Topology

- A single NURBS patch is either a topological disk, a cyclinder or a torus.

Problems of NURBS Surfaces: Topology

- A single NURBS patch is either a topological disk, a cyclinder or a torus.
- One needs to stitch several NURBS patches together to realize more complex topologies.

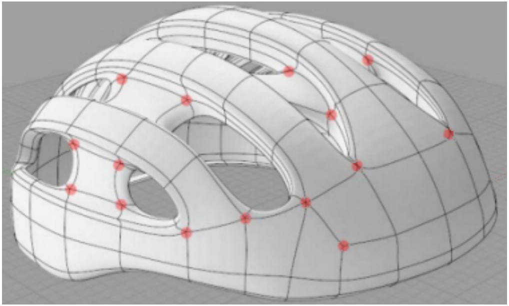
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Problems of NURBS Surfaces: Topology

- A single NURBS patch is either a topological disk, a cyclinder or a torus.
- One needs to stitch several NURBS patches together to realize more complex topologies.
- Care has to be taken at the seams of the patches to avoid cracks when such a model is deformed.

Problems of NURBS Surfaces: Topology

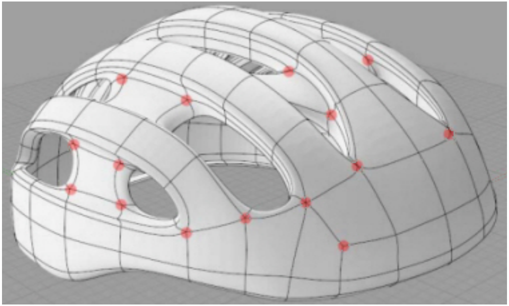
- A single NURBS patch is either a topological disk, a cyclinder or a torus.
- One needs to stitch several NURBS patches together to realize more complex topologies.
- Care has to be taken at the seams of the patches to avoid cracks when such a model is deformed.
- [Sederberg (2003)]: T-splines allow the control mesh to contain T-junctions.
- This makes it a tad easier to model complex surfaces.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Problems of NURBS Surfaces: Topology

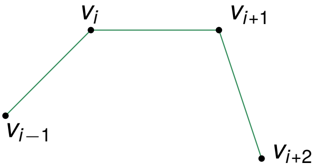
- A single NURBS patch is either a topological disk, a cyclinder or a torus.
- One needs to stitch several NURBS patches together to realize more complex topologies.
- Care has to be taken at the seams of the patches to avoid cracks when such a model is deformed.
- [Sederberg (2003)]: T-splines allow the control mesh to contain T-junctions.
- This makes it a tad easier to model complex surfaces.
- But technologies related to T-splines are patent-protected . . .
- So . . .



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Corner Cutting

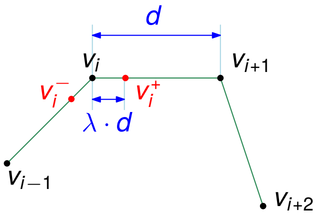
- How can we make a polygonal curve "look smooth" by manipulating its vertices?



- How can we make a polygonal curve "look smooth" by manipulating its vertices?
- [Chaikin (1974)]: Smooth the polygonal curve by iteratively replacing each vertex v_i by two new vertices v_i^- and v_i^+ such that

$$v_i^- := v_i + \lambda(v_{i-1} - v_i) \qquad \text{and} \qquad v_i^+ := v_i + \lambda(v_{i+1} - v_i)$$

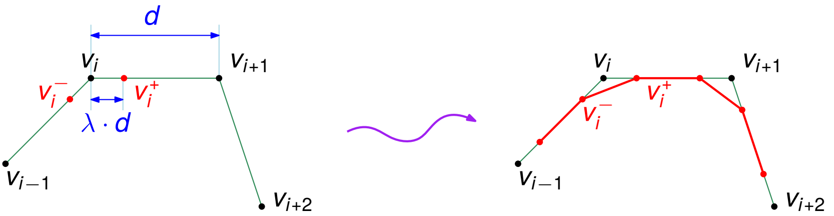
for some $\lambda \in]0, 1[$.



- How can we make a polygonal curve "look smooth" by manipulating its vertices?
- [Chaikin (1974)]: Smooth the polygonal curve by iteratively replacing each vertex v_i by two new vertices v_i^- and v_i^+ such that

$$v_i^- := v_i + \lambda(v_{i-1} - v_i) \qquad \text{and} \qquad v_i^+ := v_i + \lambda(v_{i+1} - v_i)$$

for some $\lambda \in]0, 1[$.
- Chaikin suggested $\lambda := 1/4$.

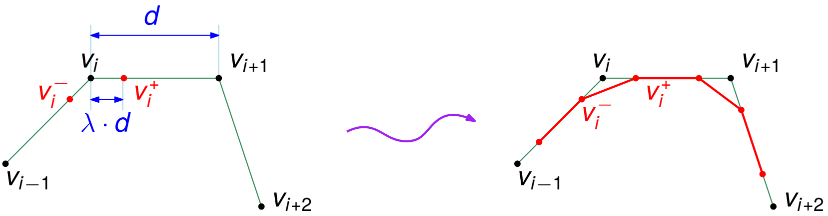


Corner Cutting

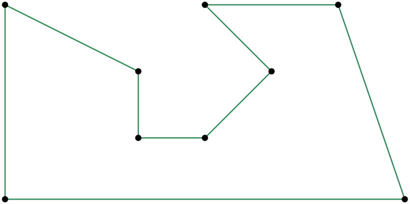
- How can we make a polygonal curve "look smooth" by manipulating its vertices?
- [Chaikin (1974)]: Smooth the polygonal curve by iteratively replacing each vertex v_i by two new vertices v_i^- and v_i^+ such that

$$v_i^- := v_i + \lambda(v_{i-1} - v_i) \quad \text{and} \quad v_i^+ := v_i + \lambda(v_{i+1} - v_i)$$

  for some $\lambda \in]0, 1[$.
- Chaikin suggested $\lambda := 1/4$.
- Need to come up with rule for handling terminal vertices.

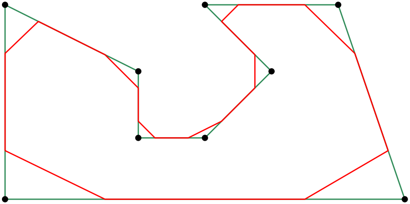


Chaikin's Corner Cutting



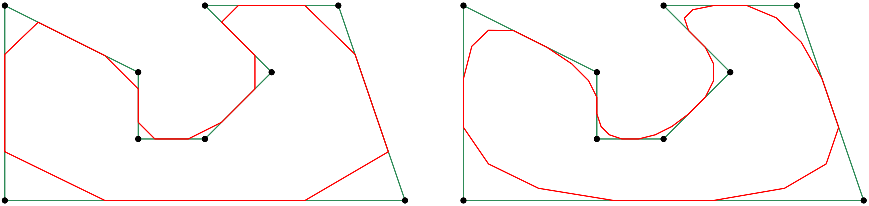
Chaikin's Corner Cutting

- Chaikin suggested $\lambda := {}^1/_4$.

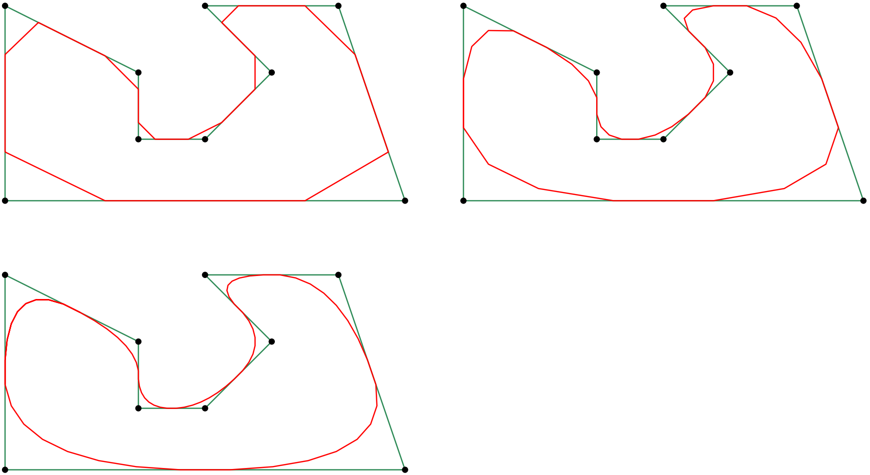


Chaikin's Corner Cutting

- Chaikin suggested $\lambda := {}^{1}\!/_{4}$. His scheme can be applied repeatedly.

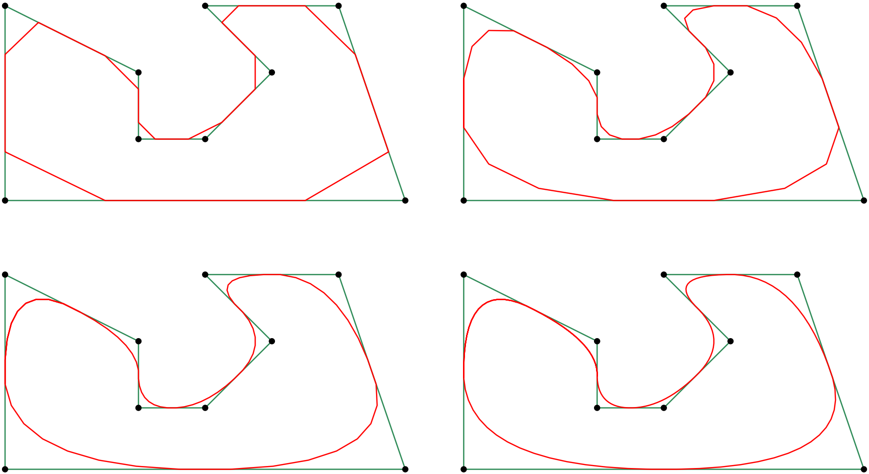


- Chaikin suggested $\lambda := 1/4$. His scheme can be applied repeatedly.



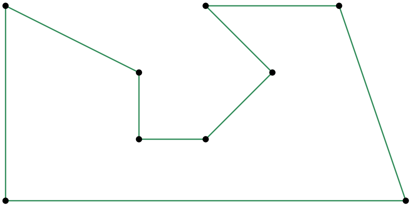
Chaikin's Corner Cutting

- Chaikin suggested $\lambda := 1/4$. His scheme can be applied repeatedly.



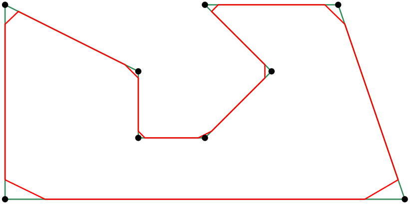
Corner Cutting

- Of course, the result depends on the value of λ.



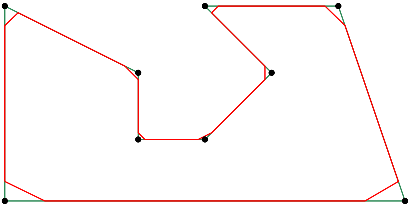
Corner Cutting

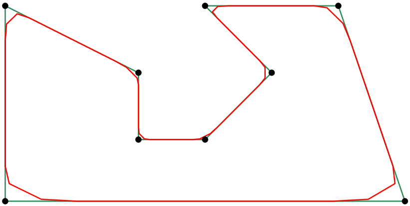
- Of course, the result depends on the value of λ. E.g., for $\lambda := {}^1\!/_{10}$:



Corner Cutting

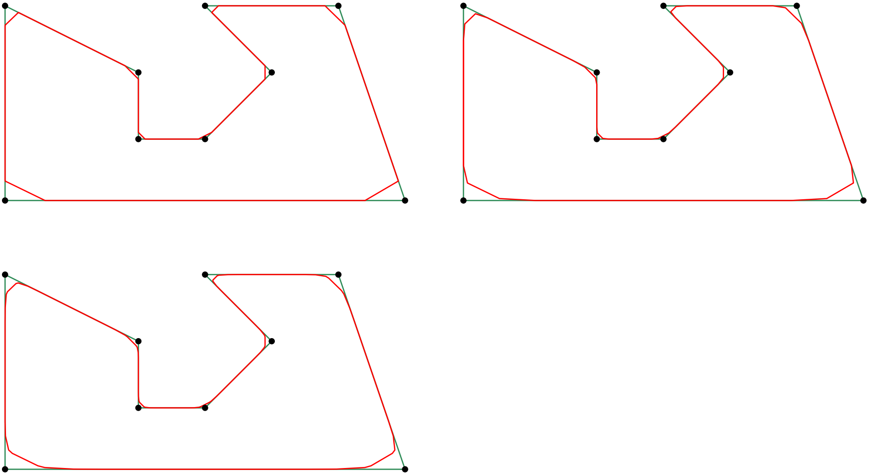
- Of course, the result depends on the value of λ. E.g., for $\lambda := {}^{1}/_{10}$:





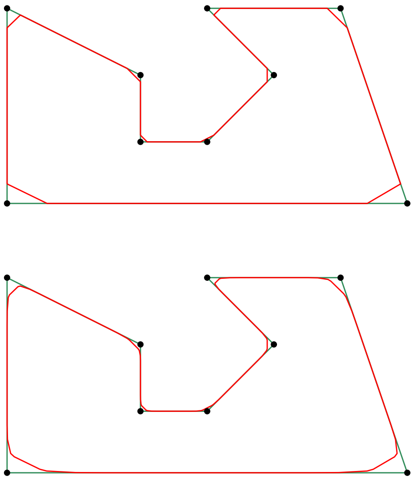
Corner Cutting

- Of course, the result depends on the value of λ. E.g., for $\lambda := {}^1\!/_{10}$:



Corner Cutting

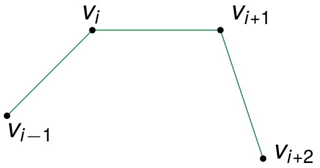
- Of course, the result depends on the value of λ. E.g., for $\lambda := {}^1\!/_{10}$:



Corner Cutting Modified

- Chaikin's corner cutting scheme replaces the vertex v_i by

$$v_i^- := \frac{3}{4} v_i + \frac{1}{4} v_{i-1} \qquad \text{and} \qquad v_i^+ := \frac{3}{4} v_i + \frac{1}{4} v_{i+1}.$$



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

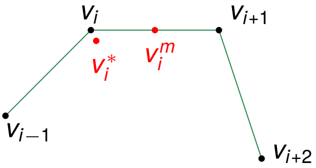
Corner Cutting Modified

- Chaikin's corner cutting scheme replaces the vertex v_i by

$$v_i^- := \frac{3}{4}v_i + \frac{1}{4}v_{i-1} \qquad \text{and} \qquad v_i^+ := \frac{3}{4}v_i + \frac{1}{4}v_{i+1}.$$

- [Catmull&Clark (1978)] modify this scheme by replacing v_i by

$$v_i^* := \frac{1}{8}v_{i-1} + \frac{3}{4}v_i + \frac{1}{8}v_{i+1} \qquad \text{and} \qquad v_i^m := \frac{1}{2}v_i + \frac{1}{2}v_{i+1}.$$



Corner Cutting Modified

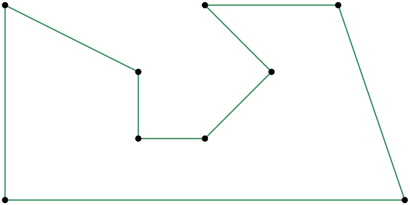
- Chaikin's corner cutting scheme replaces the vertex v_i by

$$v_i^- := \frac{3}{4}v_i + \frac{1}{4}v_{i-1} \quad \text{and} \quad v_i^+ := \frac{3}{4}v_i + \frac{1}{4}v_{i+1}.$$

- [Catmull&Clark (1978)] modify this scheme by replacing v_i by

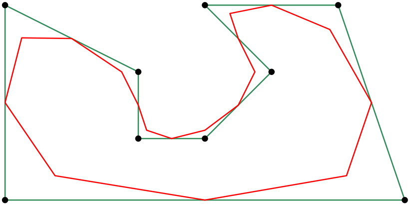
$$v_i^* := \frac{1}{8}v_{i-1} + \frac{3}{4}v_i + \frac{1}{8}v_{i+1} \quad \text{and} \quad v_i^m := \frac{1}{2}v_i + \frac{1}{2}v_{i+1}.$$

Catmull-Clark Corner Cutting



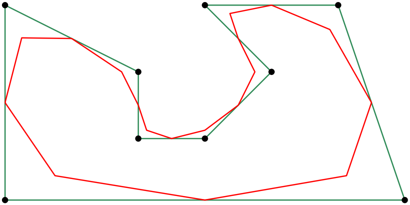
Catmull-Clark Corner Cutting

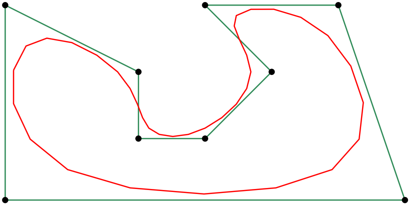
- Catmull-Clark corner cutting, with v_i^* and v_i^m replacing v_i:



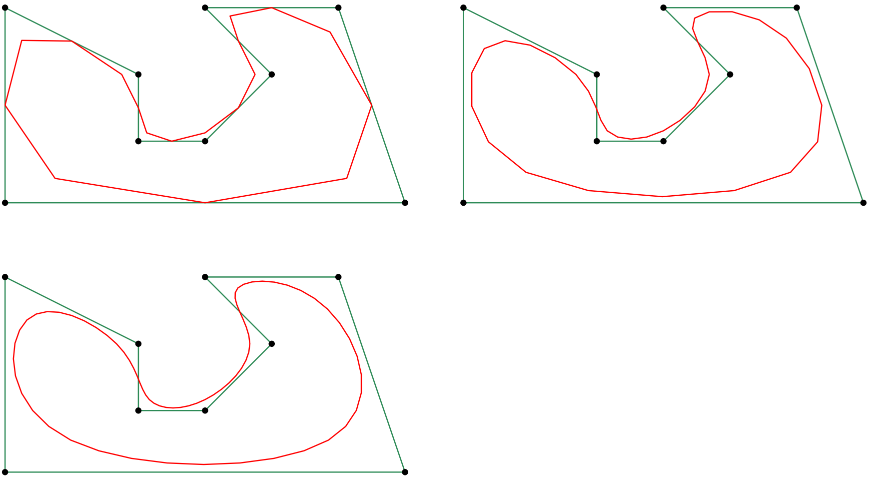
Catmull-Clark Corner Cutting

- Catmull-Clark corner cutting, with v_i^* and v_i^m replacing v_i:

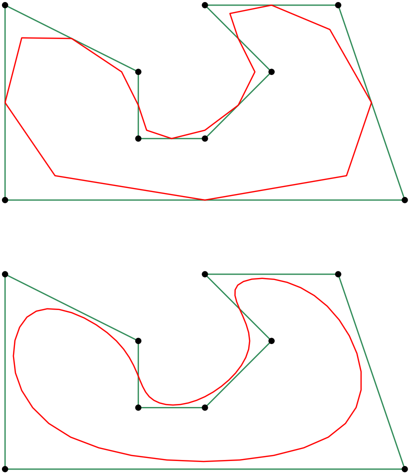




- Catmull-Clark corner cutting, with v_i^* and v_i^m replacing v_i:



- Catmull-Clark corner cutting, with v_i^* and v_i^m replacing v_i:



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

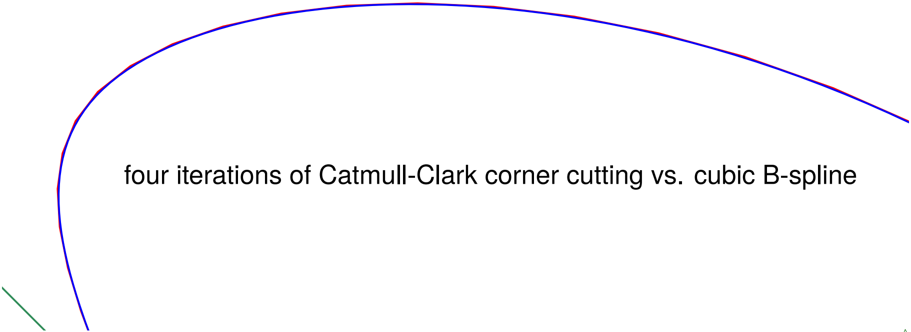
Limit Curve of Corner Cutting

Theorem 176

Chaikin's corner cutting converges to the quadratic B-spline defined by the input polygon, and Catmull-Clark corner cutting converges to its cubic B-spline.

Theorem 176

Chaikin's corner cutting converges to the quadratic B-spline defined by the input polygon, and Catmull-Clark corner cutting converges to its cubic B-spline.



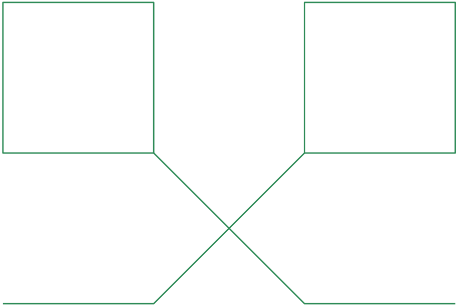
four iterations of Catmull-Clark corner cutting vs. cubic B-spline

Corner Cutting for Planar Straight-Line Graphs

- Corner cutting is based on computing weighted averages of two or three neighboring vertices.

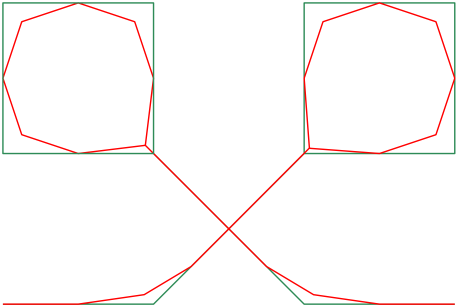
Corner Cutting for Planar Straight-Line Graphs

- Corner cutting is based on computing weighted averages of two or three neighboring vertices.
- It can be extended to arbitrary planar straight-line graphs.



Corner Cutting for Planar Straight-Line Graphs

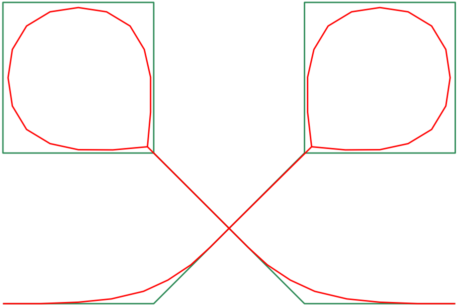
- Corner cutting is based on computing weighted averages of two or three neighboring vertices.
- It can be extended to arbitrary planar straight-line graphs.
- We get piecewise splines.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Corner Cutting for Planar Straight-Line Graphs

- Corner cutting is based on computing weighted averages of two or three neighboring vertices.
- It can be extended to arbitrary planar straight-line graphs.
- We get piecewise splines.
- That is, in the limit we get curves that are C^1-continuous or even C^2-continuous everywhere except at points that correspond to input vertices of degree three or higher.



Definition 177 (Mesh)

A *(polygon) mesh* is a collection of m plane polygons ("*faces*") such that the following conditions hold:

1. Every pair of polygons intersects at most in common edges or common vertices.
2. The union of all m polygons forms (part of) a 2-manifold.

Definition 177 (Mesh)

A *(polygon) mesh* is a collection of m plane polygons ("*faces*") such that the following conditions hold:

1. Every pair of polygons intersects at most in common edges or common vertices.
2. The union of all m polygons forms (part of) a 2-manifold.

A mesh is *closed* if every edge is shared by exactly two polygons. Otherwise, it is *open* and has *boundary edges*.

Definition 177 (Mesh)

A *(polygon) mesh* is a collection of m plane polygons ("*faces*") such that the following conditions hold:

1. Every pair of polygons intersects at most in common edges or common vertices.
2. The union of all m polygons forms (part of) a 2-manifold.

A mesh is *closed* if every edge is shared by exactly two polygons. Otherwise, it is *open* and has *boundary edges*.
The *degree of a vertex* of a mesh is given by the number of incident polygon edges.

Definition 177 (Mesh)

A *(polygon) mesh* is a collection of m plane polygons ("*faces*") such that the following conditions hold:

1. Every pair of polygons intersects at most in common edges or common vertices.
2. The union of all m polygons forms (part of) a 2-manifold.

A mesh is *closed* if every edge is shared by exactly two polygons. Otherwise, it is *open* and has *boundary edges*.
The *degree of a vertex* of a mesh is given by the number of incident polygon edges.

- The surface of a polyhedron forms a closed mesh.

Mesh

Definition 177 (Mesh)

A *(polygon) mesh* is a collection of m plane polygons ("*faces*") such that the following conditions hold:

1. Every pair of polygons intersects at most in common edges or common vertices.
2. The union of all m polygons forms (part of) a 2-manifold.

A mesh is *closed* if every edge is shared by exactly two polygons. Otherwise, it is *open* and has *boundary edges*.
The *degree of a vertex* of a mesh is given by the number of incident polygon edges.

- The surface of a polyhedron forms a closed mesh.
- Typical faces are given by triangles and plane quads.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 177 (Mesh)

A *(polygon) mesh* is a collection of m plane polygons ("*faces*") such that the following conditions hold:

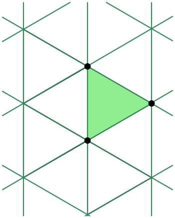
1. Every pair of polygons intersects at most in common edges or common vertices.
2. The union of all m polygons forms (part of) a 2-manifold.

A mesh is *closed* if every edge is shared by exactly two polygons. Otherwise, it is *open* and has *boundary edges*.
The *degree of a vertex* of a mesh is given by the number of incident polygon edges.

- The surface of a polyhedron forms a closed mesh.
- Typical faces are given by triangles and plane quads.
- Recall that Euler's formula $v - e + f = 2$ is applicable to closed (connected) meshes.

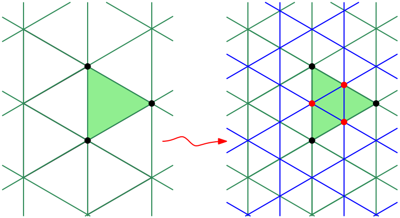
Mesh Subdivision: Face Split versus Vertex Split

- Smaller faces of a mesh can be generated by



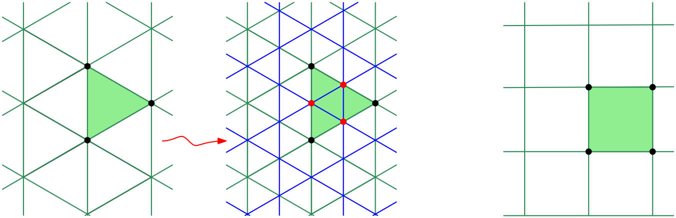
Mesh Subdivision: Face Split versus Vertex Split

- Smaller faces of a mesh can be generated by
  1. splitting a face into sub-faces



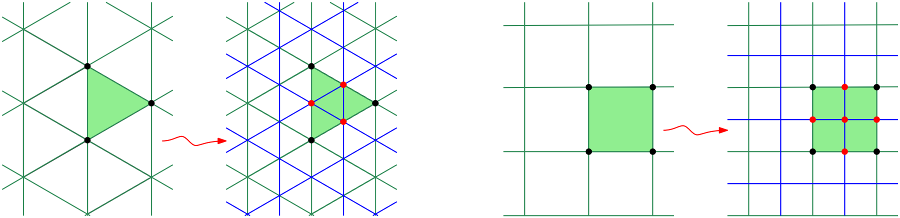
Mesh Subdivision: Face Split versus Vertex Split

- Smaller faces of a mesh can be generated by
  1. splitting a face into sub-faces



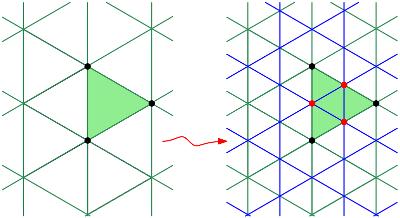
Mesh Subdivision: Face Split versus Vertex Split

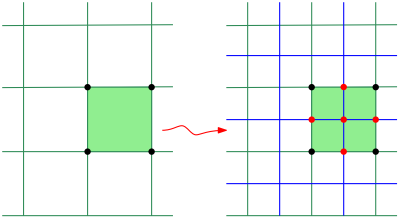
- Smaller faces of a mesh can be generated by
  1. splitting a face into sub-faces

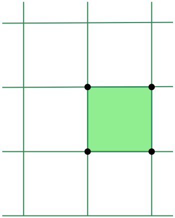


Mesh Subdivision: Face Split versus Vertex Split

- Smaller faces of a mesh can be generated by
  1. splitting a face into sub-faces, and/or
  2. splitting a vertex.

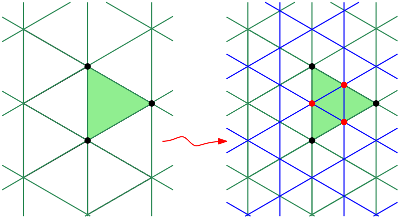


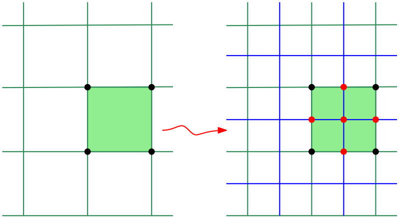


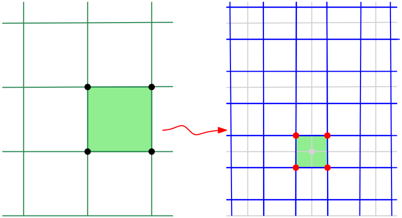


Mesh Subdivision: Face Split versus Vertex Split

- Smaller faces of a mesh can be generated by
  1. splitting a face into sub-faces, and/or
  2. splitting a vertex.







UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Mesh Subdivision: Goals

Efficiency: The computation of the positions of the new vertices should be efficient, based on only a small number of arithmetic operations.

Simplicity: A small number of simple subdivision rules is sought.

Mesh Subdivision: Goals

Efficiency: The computation of the positions of the new vertices should be efficient, based on only a small number of arithmetic operations.

Simplicity: A small number of simple subdivision rules is sought.

Local control: The subdivision rules that define a new point should involve only points that are “close by”.

Local support: The position of an input point influences only a small area of the final shape.

Mesh Subdivision: Goals

Efficiency: The computation of the positions of the new vertices should be efficient, based on only a small number of arithmetic operations.

Simplicity: A small number of simple subdivision rules is sought.

Local control: The subdivision rules that define a new point should involve only points that are "close by".

Local support: The position of an input point influences only a small area of the final shape.

Affine invariance: An affine transformation applied to the vertices of the original mesh followed by some subdivision steps should define the same surface as obtained by transforming the shape given after some subdivision steps relative to the original mesh.

Mesh Subdivision: Goals

Efficiency: The computation of the positions of the new vertices should be efficient, based on only a small number of arithmetic operations.

Simplicity: A small number of simple subdivision rules is sought.

Local control: The subdivision rules that define a new point should involve only points that are "close by".

Local support: The position of an input point influences only a small area of the final shape.

Affine invariance: An affine transformation applied to the vertices of the original mesh followed by some subdivision steps should define the same surface as obtained by transforming the shape given after some subdivision steps relative to the original mesh.

Smoothness: The limit curve/surface should be of provable continuity.

Mesh Subdivision: Goals

Efficiency: The computation of the positions of the new vertices should be efficient, based on only a small number of arithmetic operations.

Simplicity: A small number of simple subdivision rules is sought.

Local control: The subdivision rules that define a new point should involve only points that are "close by".

Local support: The position of an input point influences only a small area of the final shape.

Affine invariance: An affine transformation applied to the vertices of the original mesh followed by some subdivision steps should define the same surface as obtained by transforming the shape given after some subdivision steps relative to the original mesh.

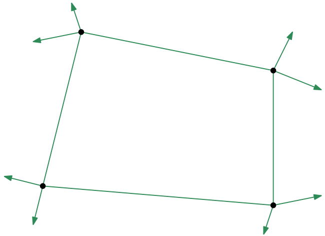
Smoothness: The limit curve/surface should be of provable continuity.

Special surface features: Creases, grooves and sharp points/edges should be representable.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Subdivision for Regular Quad Meshes

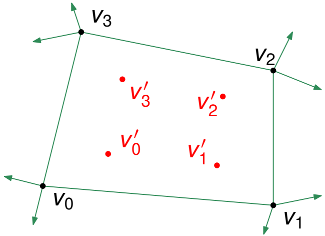
- Suppose that all faces in a mesh are quadrilaterals and that every vertex is shared by exactly four faces.



Subdivision for Regular Quad Meshes

- Suppose that all faces in a mesh are quadrilaterals and that every vertex is shared by exactly four faces.
- For each of the four vertices v_0, v_1, v_2, v_3 of a quadrilateral, four new vertices v_0', v_1', v_2', v_3' are computed as follows (with indices taken modulo four):

$$v_i' := \frac{3}{16} v_{i-1} + \frac{9}{16} v_i + \frac{3}{16} v_{i+1} + \frac{1}{16} v_{i+2}$$

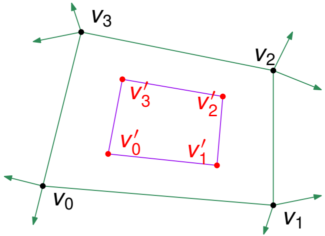


Subdivision for Regular Quad Meshes

- Suppose that all faces in a mesh are quadrilaterals and that every vertex is shared by exactly four faces.
- For each of the four vertices v_0, v_1, v_2, v_3 of a quadrilateral, four new vertices v'_0, v'_1, v'_2, v'_3 are computed as follows (with indices taken modulo four):

$$v'_i := \frac{3}{16} v_{i-1} + \frac{9}{16} v_i + \frac{3}{16} v_{i+1} + \frac{1}{16} v_{i+2}$$

- The vertices v'_0, v'_1, v'_2, v'_3 define a new quadrilateral.

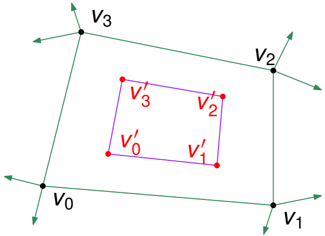


Subdivision for Regular Quad Meshes

- Suppose that all faces in a mesh are quadrilaterals and that every vertex is shared by exactly four faces.
- For each of the four vertices v_0, v_1, v_2, v_3 of a quadrilateral, four new vertices v_0', v_1', v_2', v_3' are computed as follows (with indices taken modulo four):

$$v_i' := \frac{3}{16} v_{i-1} + \frac{9}{16} v_i + \frac{3}{16} v_{i+1} + \frac{1}{16} v_{i+2}$$

- The vertices v_0', v_1', v_2', v_3' define a new quadrilateral.
- Similarly for the new vertices in the other quadrilaterals.



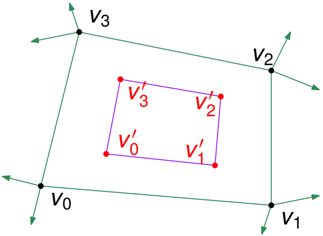
Doo-Sabin Subdivision: Weighted Averages

- [Doo&Sabin (1978)]: The new vertices $v'_1, v'_2, \ldots, v'_k$ of a face with k vertices are obtained as follows (for $1 \leq i \leq k$):

$$v'_i := \sum_{j=1}^{k} \alpha_{ij} v_j,$$

where

$$a_{ij} := \begin{cases} \frac{k+5}{4k} & \text{if } i = j, \\ \frac{1}{4k}\left[3 + 2\cos\left(\frac{2\pi(i-j)}{k}\right)\right] & \text{otherwise.} \end{cases}$$



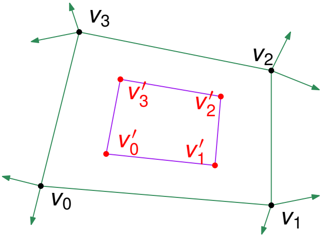
- [Doo&Sabin (1978)]: The new vertices $v_1', v_2', \ldots, v_k'$ of a face with k vertices are obtained as follows (for $1 \le i \le k$):

$$v_i' := \sum_{j=1}^{k} \alpha_{ij} v_j,$$

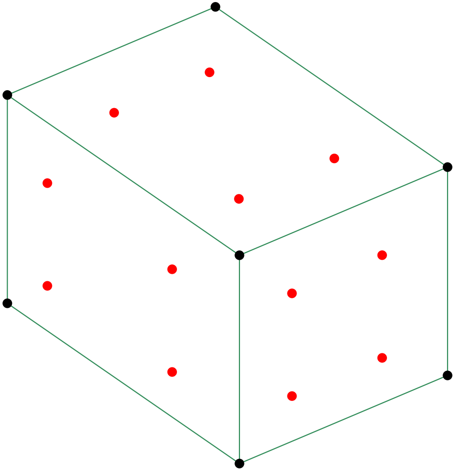
where

$$a_{ij} := \begin{cases} \frac{k+5}{4k} & \text{if } i = j, \\ \frac{1}{4k}\left[3 + 2\cos\left(\frac{2\pi(i-j)}{k}\right)\right] & \text{otherwise.} \end{cases}$$

- Note that this formula matches the formula given for quads on the previous slide!

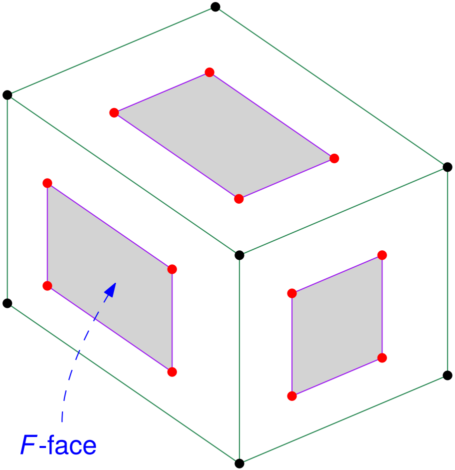


Doo-Sabin Subdivision: Remeshing



- Remeshing the new face vertices yields three types of faces.

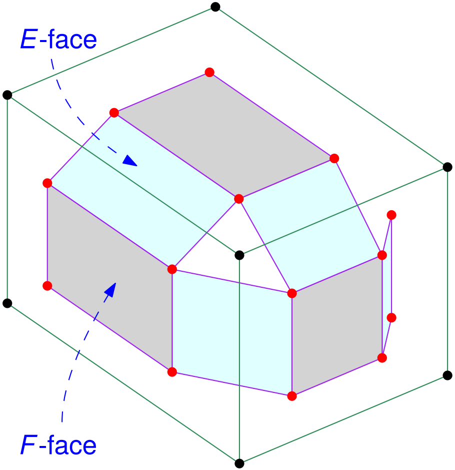
Doo-Sabin Subdivision: Remeshing



- Remeshing the new face vertices yields three types of faces.
- An F-Face is defined by the new vertices of one face. It replaces the old face.

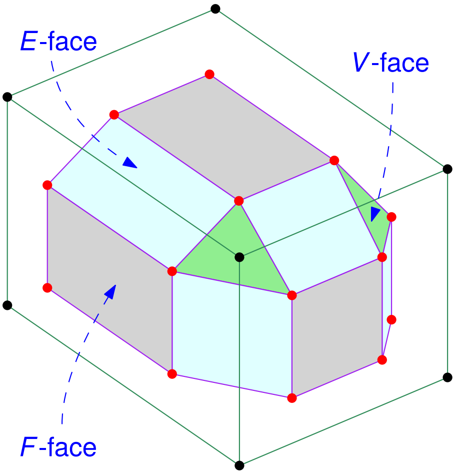
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Doo-Sabin Subdivision: Remeshing



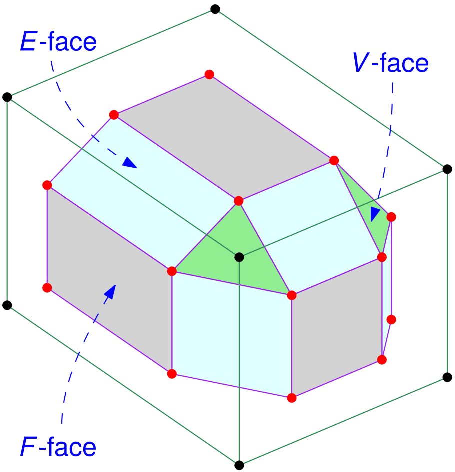
- Remeshing the new face vertices yields three types of faces.
- An *F*-Face is defined by the new vertices of one face. It replaces the old face.
- An *E*-face corresponds to an old edge.

Doo-Sabin Subdivision: Remeshing



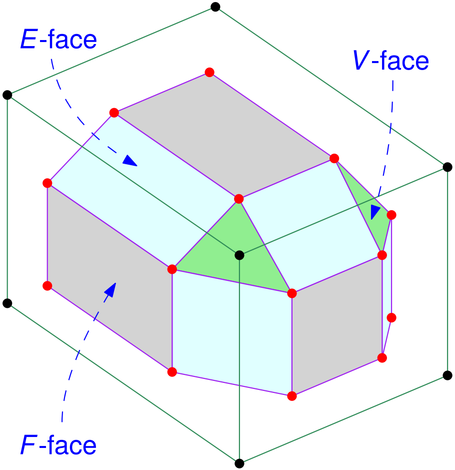
- Remeshing the new face vertices yields three types of faces.
- An *F*-Face is defined by the new vertices of one face. It replaces the old face.
- An *E*-face corresponds to an old edge.
- An *V*-face corresponds to an old vertex.

Doo-Sabin Subdivision: Remeshing



- Remeshing the new face vertices yields three types of faces.
- An *F*-Face is defined by the new vertices of one face. It replaces the old face.
- An *E*-face corresponds to an old edge.
- An *V*-face corresponds to an old vertex.
- If the input mesh consists of quadrilaterals then most new faces are quadrilaterals, too.

Doo-Sabin Subdivision: Remeshing



- Remeshing the new face vertices yields three types of faces.
- An *F*-Face is defined by the new vertices of one face. It replaces the old face.
- An *E*-face corresponds to an old edge.
- An *V*-face corresponds to an old vertex.
- If the input mesh consists of quadrilaterals then most new faces are quadrilaterals, too.
- Non four-sided new faces are *V*-faces that correspond to "extraordinary" vertices whose degree is not four.

Doo-Sabin Subdivision: Properties

- After one round of subdivision, all vertices are of degree four.

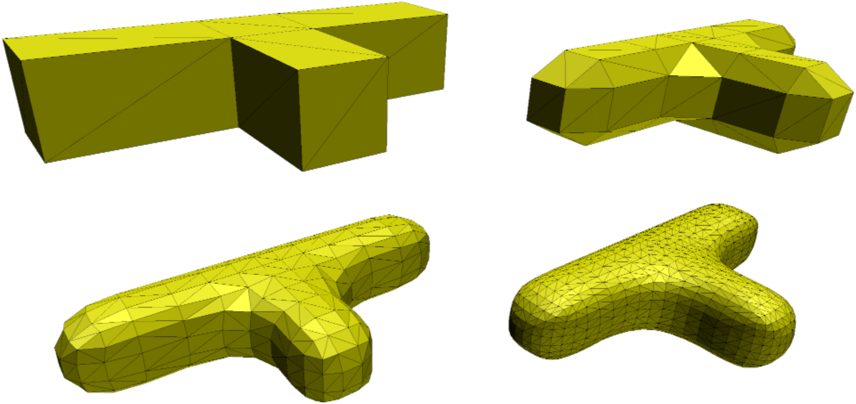
Doo-Sabin Subdivision: Properties

- After one round of subdivision, all vertices are of degree four.

Lemma 178

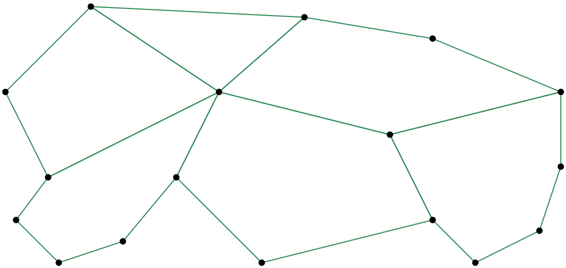
The limit surface of Doo-Sabin subdivision mostly is a B-spline surface of degree (2,2). It is C^1 everywhere except at points that correspond to extraordinary vertices where it is only G^1.

Doo-Sabin Subdivision: Sample



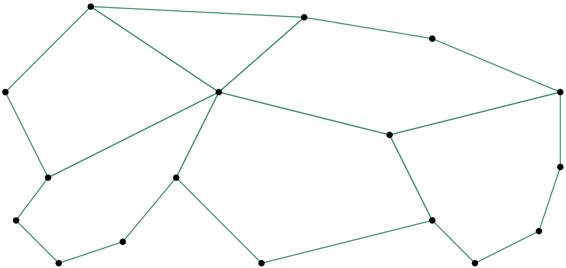
Catmull-Clark Subdivision: Unweighted Averages

- [Catmull&Clark (1978)]: They compute a face point for every face, followed by an edge point for every edge, and then a vertex point for every vertex.



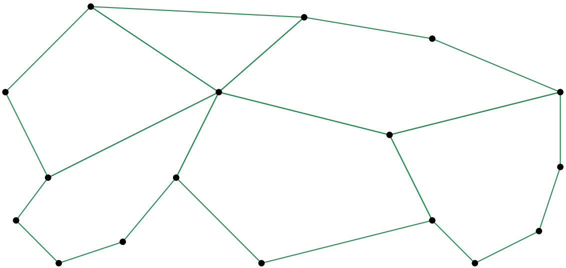
Catmull-Clark Subdivision: Unweighted Averages

- [Catmull&Clark (1978)]: They compute a face point for every face, followed by an edge point for every edge, and then a vertex point for every vertex.
- Once these new vertices are available, a new mesh is constructed.



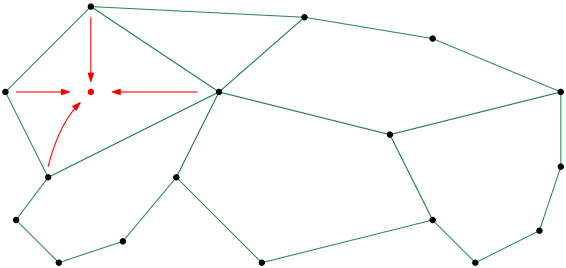
Catmull-Clark Subdivision: Unweighted Averages

- [Catmull&Clark (1978)]: They compute a face point for every face, followed by an edge point for every edge, and then a vertex point for every vertex.
- Once these new vertices are available, a new mesh is constructed.
- We assume that the surface is a 2-manifold without boundary.



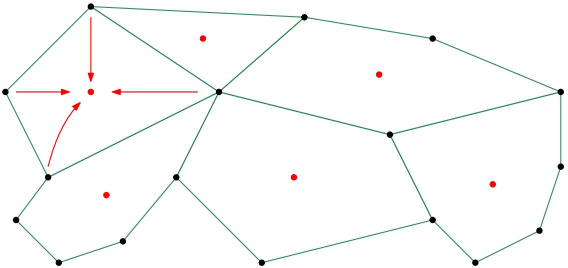
Catmull-Clark Subdivision: Unweighted Averages

- [Catmull&Clark (1978)]: They compute a face point for every face, followed by an edge point for every edge, and then a vertex point for every vertex.
- Once these new vertices are available, a new mesh is constructed.
- We assume that the surface is a 2-manifold without boundary.
- A face point is given by the centroid of that face, i.e., by the average of its vertices.



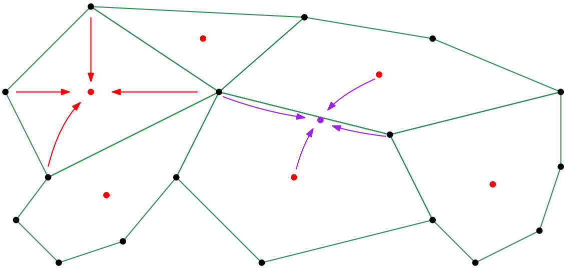
Catmull-Clark Subdivision: Unweighted Averages

- [Catmull&Clark (1978)]: They compute a face point for every face, followed by an edge point for every edge, and then a vertex point for every vertex.
- Once these new vertices are available, a new mesh is constructed.
- We assume that the surface is a 2-manifold without boundary.
- A face point is given by the centroid of that face, i.e., by the average of its vertices.

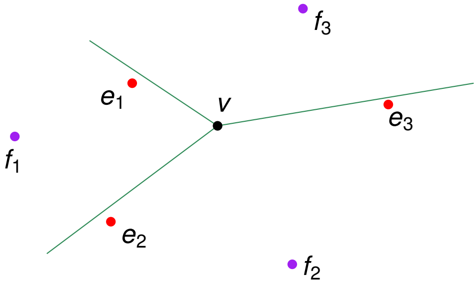


Catmull-Clark Subdivision: Unweighted Averages

- [Catmull&Clark (1978)]: They compute a face point for every face, followed by an edge point for every edge, and then a vertex point for every vertex.
- Once these new vertices are available, a new mesh is constructed.
- We assume that the surface is a 2-manifold without boundary.
- A face point is given by the centroid of that face, i.e., by the average of its vertices.
- An edge point is given by the average of the two end-points of that edge and the face points of its two adjacent faces.



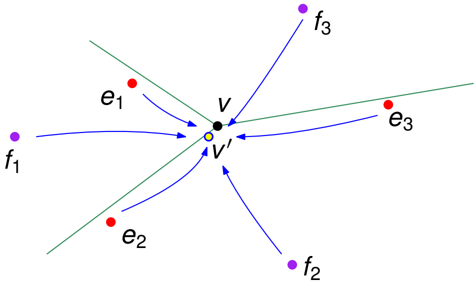
- Let $e_1, e_2, \dots, e_k$ and $f_1, f_2, \dots, f_k$ be the edge and face points of the k edges (resp., faces) incident at a vertex v.



Catmull-Clark Subdivision: Weighted Averages

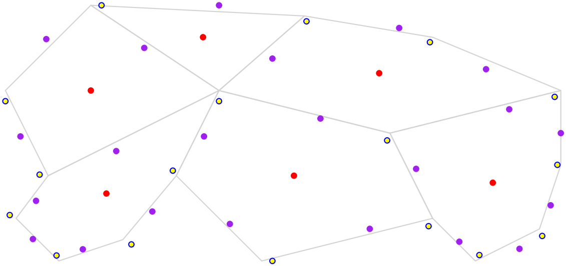
- Let $e_1, e_2, \ldots, e_k$ and $f_1, f_2, \ldots, f_k$ be the edge and face points of the k edges (resp., faces) incident at a vertex v.
- The position v' of the relocated vertex point is computed as follows:

$$v' = \frac{k-3}{k} v + \frac{1}{k} \sum_{i=1}^{k} f_i + \frac{2}{k} \sum_{i=1}^{k} e_i$$

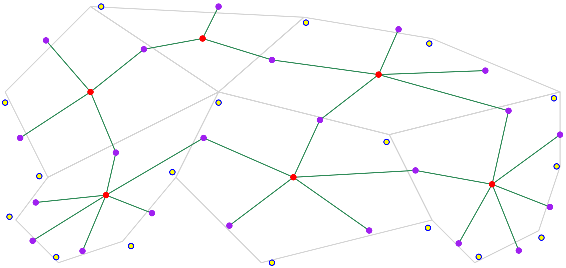


UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Catmull-Clark Subdivision: Remeshing

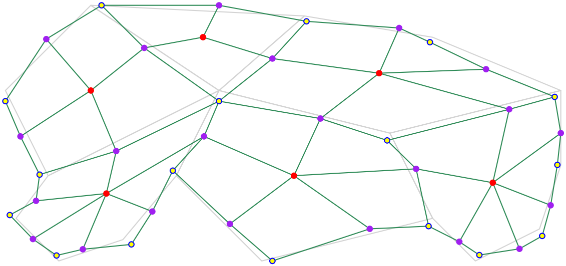


- Connect every face point to the edge points of its edges.



Catmull-Clark Subdivision: Remeshing

- Connect every face point to the edge points of its edges.
- Connect every vertex point to the edge points of the edges incident to it.



Catmull-Clark Subdivision: Properties

Lemma 179

All faces of the mesh are quadrilaterals after one run of Catmull-Clark subdivision.

Catmull-Clark Subdivision: Properties

Lemma 179

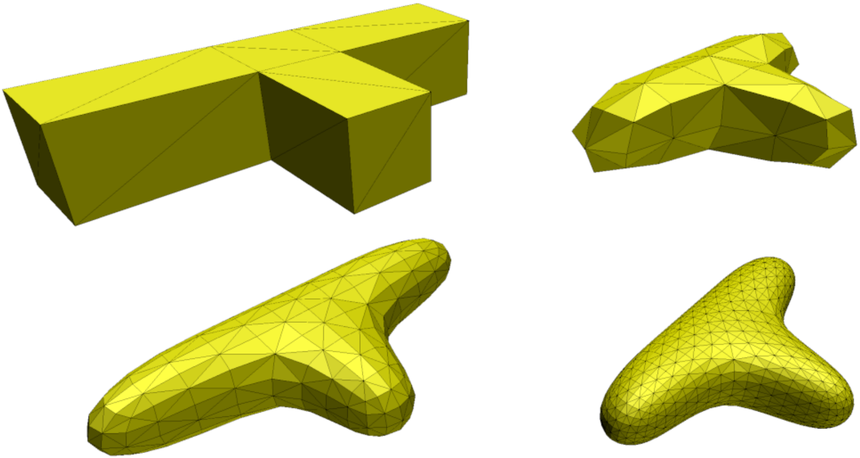
All faces of the mesh are quadrilaterals after one run of Catmull-Clark subdivision.

- For Catmull-Clark subdivision, a vertex is extraordinary if it is not of degree four.

Lemma 180

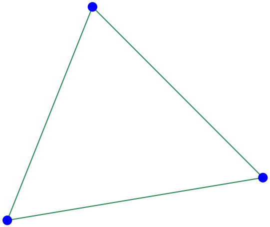
The limit surface of Catmull-Clark subdivision is a B-spline surface of degree $(3, 3)$. It is C^2 everywhere except at points that correspond to extraordinary vertices where it is only C^1.

Catmull-Clark Subdivision: Sample



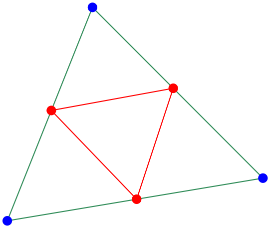
Loop Subdivision: Even and Odd Vertices

- Consider a mesh with only triangular faces.



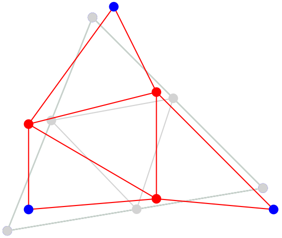
Loop Subdivision: Even and Odd Vertices

- Consider a mesh with only triangular faces.
- [Loop (1987)]:
  1. Split every triangle into three sub-triangles by inserting three new vertices on its edges ("edge points") .



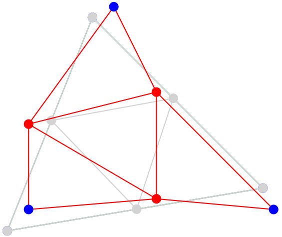
Loop Subdivision: Even and Odd Vertices

- Consider a mesh with only triangular faces.
- [Loop (1987)]:
  1. Split every triangle into three sub-triangles by inserting three new vertices on its edges (“edge points”) .
  2. Relocate old and new vertices by computing weighted averages.



Loop Subdivision: Even and Odd Vertices

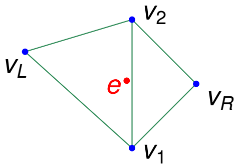
- Consider a mesh with only triangular faces.
- [Loop (1987)]:
  1. Split every triangle into three sub-triangles by inserting three new vertices on its edges (“edge points”) .
  2. Relocate old and new vertices by computing weighted averages.
- Old vertices are commonly called *even* vertices, and new vertices are called *odd* vertices.



Loop Subdivision: Weighted Averages

- For an edge point e defined by a non-boundary edge $\overline{v_1 v_2}$:

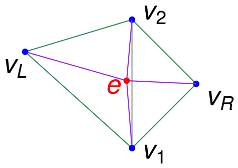
$$e := \frac{3}{8}(v_1 + v_2) + \frac{1}{8}(v_L + v_R)$$



Loop Subdivision: Weighted Averages

- For an edge point e defined by a non-boundary edge $\overline{v_1 v_2}$:

$$e := \frac{3}{8}(v_1 + v_2) + \frac{1}{8}(v_L + v_R)$$



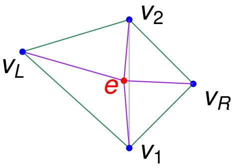
Loop Subdivision: Weighted Averages

- For an edge point e defined by a non-boundary edge $\overline{v_1 v_2}$:

$$e := \frac{3}{8}(v_1 + v_2) + \frac{1}{8}(v_L + v_R)$$

- For an edge point e defined by a boundary edge $\overline{v_1 v_2}$:

$$e := \frac{1}{2}(v_1 + v_2)$$

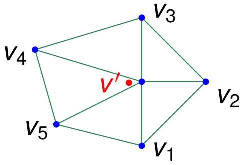


- The new position v' of a non-boundary vertex v with k old neighbors $v_1, v_2, \ldots, v_k$ is computed as

$$v' := (1-\alpha)v + \alpha\bar{v}$$

where

$$\bar{v} := \frac{1}{k}\sum_{j=1}^{k} v_k \qquad \text{and} \qquad \alpha := \begin{cases} \frac{3}{16} & \text{if } k = 3, \\ \frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2 & \text{if } k > 3. \end{cases}$$

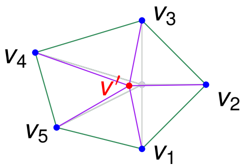


- The new position v' of a non-boundary vertex v with k old neighbors $v_1, v_2, \ldots, v_k$ is computed as

$$v' := (1-\alpha)v + \alpha\bar{v}$$

where

$$\bar{v} := \frac{1}{k}\sum_{j=1}^{k} v_k \qquad \text{and} \qquad \alpha := \begin{cases} \frac{3}{16} & \text{if } k = 3, \\ \frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2 & \text{if } k > 3. \end{cases}$$



Loop Subdivision: Weighted Averages

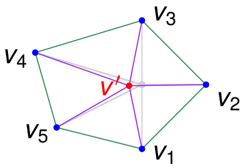
- The new position v' of a non-boundary vertex v with k old neighbors $v_1, v_2, \ldots, v_k$ is computed as

$$v' := (1-\alpha)v + \alpha\bar{v}$$

where

$$\bar{v} := \frac{1}{k}\sum_{j=1}^{k} v_k \qquad \text{and} \qquad \alpha := \begin{cases} \frac{3}{16} & \text{if } k = 3, \\ \frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2 & \text{if } k > 3. \end{cases}$$

- [Warren (1995)]: Use $\alpha := \frac{3}{8k}$ for $k > 3$.



Loop Subdivision: Weighted Averages

- The new position v' of a non-boundary vertex v with k old neighbors $v_1, v_2, \ldots, v_k$ is computed as

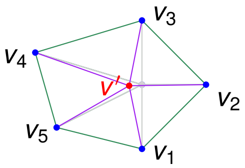
$$v' := (1-\alpha)v + \alpha\bar{v}$$

where

$$\bar{v} := \frac{1}{k}\sum_{j=1}^{k} v_k \qquad \text{and} \qquad \alpha := \begin{cases} \frac{3}{16} & \text{if } k = 3, \\ \frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2 & \text{if } k > 3. \end{cases}$$

- [Warren (1995)]: Use $\alpha := \frac{3}{8k}$ for $k > 3$.
- A boundary vertex v is relocated as

$$v' := \frac{1}{8}(v_L + v_R) + \frac{3}{4}v.$$



Loop Subdivision: Properties

Lemma 181

A Loop subdivision surface lies within the convex hull of its input vertices.

Lemma 181

A Loop subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 181

A Loop subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

- For Loop subdivision, a vertex is called *extraordinary* if its degree is not equal to six.

Lemma 181

A Loop subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

- For Loop subdivision, a vertex is called *extraordinary* if its degree is not equal to six.

Lemma 182

The limit surface of Loop subdivision is a generalization of box splines. It is C^2 except for extraordinary vertices where it is only G^1. Same for Warren's simplification.

Lemma 181

A Loop subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

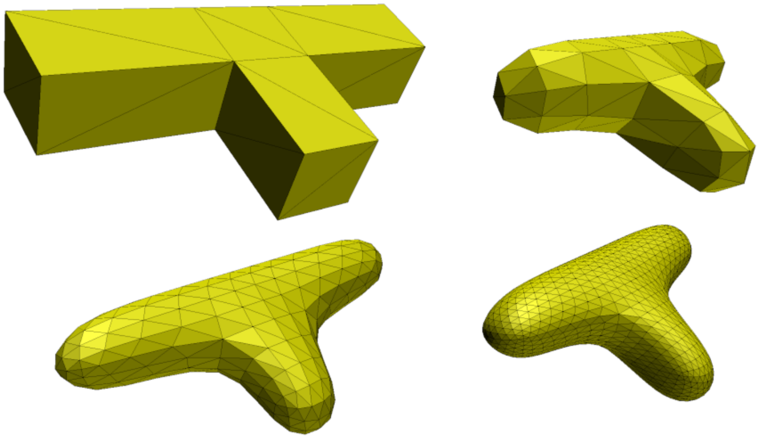
- For Loop subdivision, a vertex is called *extraordinary* if its degree is not equal to six.

Lemma 182

The limit surface of Loop subdivision is a generalization of box splines. It is C^2 except for extraordinary vertices where it is only G^1. Same for Warren's simplification.

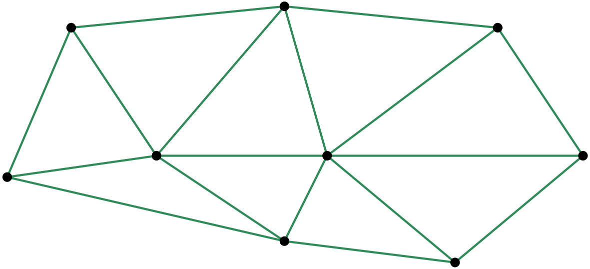
- For meshes with non-triangular faces the final limit surface depends on the triangulation of those faces.

Loop Subdivision: Sample

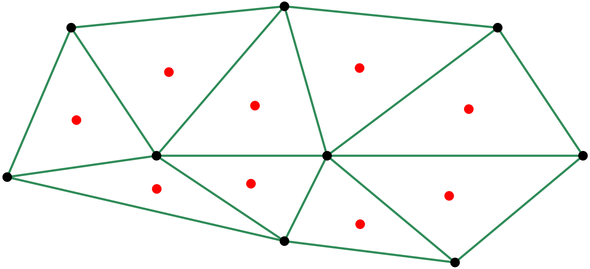


$\sqrt{3}$ Subdivision

- Consider a mesh with only triangular faces.

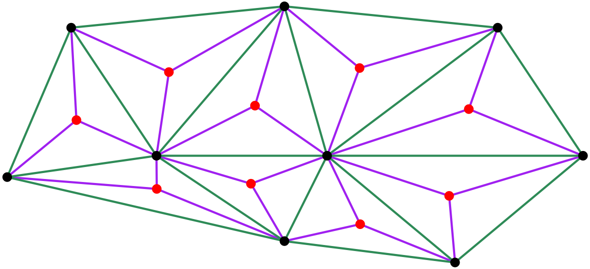


- Consider a mesh with only triangular faces.
- Kobbelt (2000)]:
  1. Split every triangle into three sub-triangles by inserting a center vertex at the centroid of each triangle.

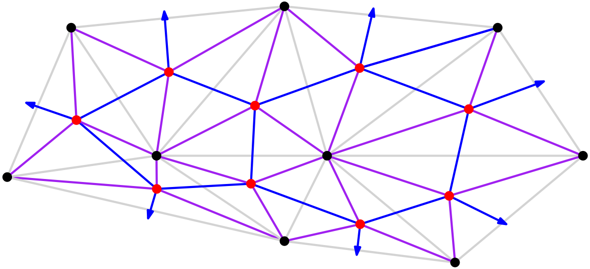


$\sqrt{3}$ Subdivision

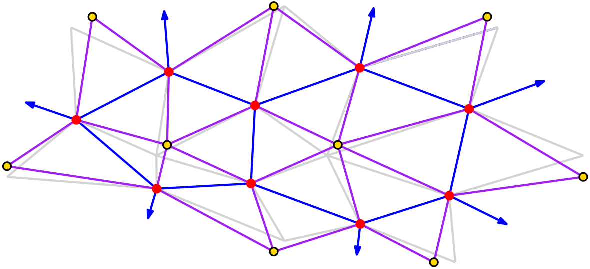
- Consider a mesh with only triangular faces.
- Kobbelt (2000)]:
  1. Split every triangle into three sub-triangles by inserting a center vertex at the centroid of each triangle.



- Consider a mesh with only triangular faces.
- Kobbelt (2000)]:
  1. Split every triangle into three sub-triangles by inserting a center vertex at the centroid of each triangle.
  2. Flip all original triangle edges. This yields a new triangular mesh.

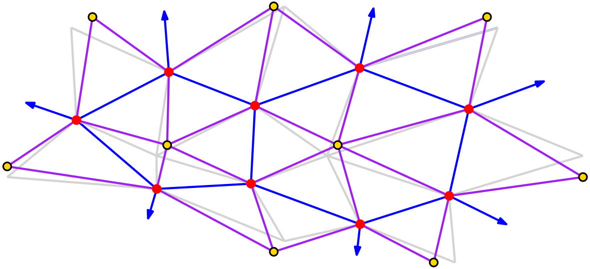


- Consider a mesh with only triangular faces.
- Kobbelt (2000)]:
  1. Split every triangle into three sub-triangles by inserting a center vertex at the centroid of each triangle.
  2. Flip all original triangle edges. This yields a new triangular mesh.
  3. Relocate every old vertex.



- Consider a mesh with only triangular faces.
- Kobbelt (2000)]:
  1. Split every triangle into three sub-triangles by inserting a center vertex at the centroid of each triangle.
  2. Flip all original triangle edges. This yields a new triangular mesh.
  3. Relocate every old vertex.
- The centroid c of a triangle $\Delta(v_1, v_2, v_3)$ is its center of gravity:

$$c := \frac{1}{3}(v_1 + v_2 + v_3).$$

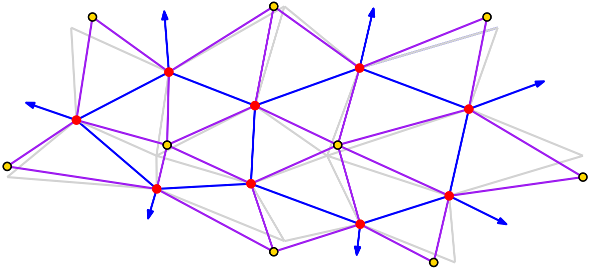


- An old vertex v with k neighbors $v_1, v_2, \ldots, v_k$ is relocated to its new position v' as follows:

$$v' := (1-\alpha)v + \alpha\bar{v},$$

with

$$\alpha := \frac{1}{9}\left[4 - 2\cos\left(\frac{2\pi}{k}\right)\right] \qquad \text{and} \qquad \bar{v} := \frac{1}{k}\sum_{j=1}^{k} v_j.$$



$\sqrt{3}$ Subdivision: Properties

Lemma 183

A $\sqrt{3}$ subdivision surface lies within the convex hull of its input vertices.

Lemma 183

A $\sqrt{3}$ subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

Lemma 183

A $\sqrt{3}$ subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

- After two subdivison steps the number of triangles has increased by a multiplicative factor of nine. (This fact motivated the name of the scheme.)

$\sqrt{3}$ Subdivision: Properties

Lemma 183

A $\sqrt{3}$ subdivision surface lies within the convex hull of its input vertices.

Sketch of proof: Note that all weights are non-negative and sum up to one. □

- After two subdivison steps the number of triangles has increased by a multiplicative factor of nine. (This fact motivated the name of the scheme.)
- For $\sqrt{3}$ subdivision, a vertex is called *extraordinary* if its degree is not equal to six.

Lemma 183

A $\sqrt{3}$ subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

- After two subdivison steps the number of triangles has increased by a multiplicative factor of nine. (This fact motivated the name of the scheme.)
- For $\sqrt{3}$ subdivision, a vertex is called *extraordinary* if its degree is not equal to six.

Lemma 184

The limit surface of $\sqrt{3}$ subdivision is a collection of C^2 patches except for extraordinary vertices where it is only C^1.

$\sqrt{3}$ Subdivision: Properties

Lemma 183

A $\sqrt{3}$ subdivision surface lies within the convex hull of its input vertices.

Sketch of proof : Note that all weights are non-negative and sum up to one. □

- After two subdivison steps the number of triangles has increased by a multiplicative factor of nine. (This fact motivated the name of the scheme.)
- For $\sqrt{3}$ subdivision, a vertex is called *extraordinary* if its degree is not equal to six.

Lemma 184

The limit surface of $\sqrt{3}$ subdivision is a collection of C^2 patches except for extraordinary vertices where it is only C^1.

- Kobbelt's $\sqrt{3}$'s scheme can be extended to an adaptive scheme for even finer control of the subdivision.

Discussion

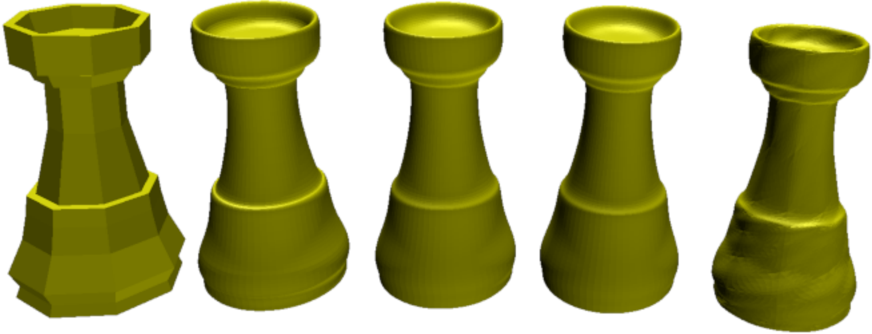
- Catmull-Clark subdivision is best for quads, and poor on triangular meshes.

Discussion

- Catmull-Clark subdivision is best for quads, and poor on triangular meshes.
- Loop subdivision and $\sqrt{3}$ subdivision work nicely for triangular meshes.

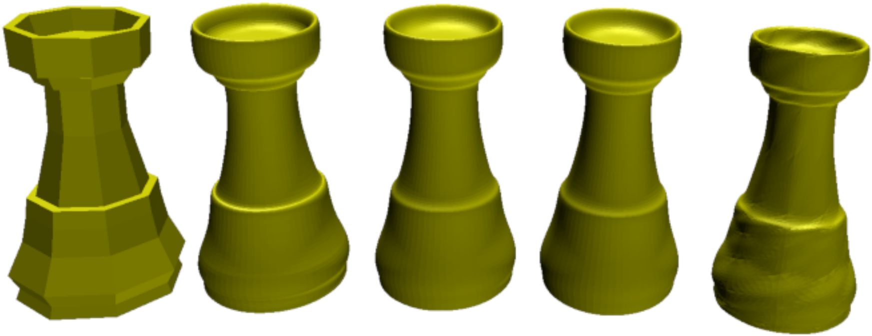
Discussion

- Catmull-Clark subdivision is best for quads, and poor on triangular meshes.
- Loop subdivision and $\sqrt{3}$ subdivision work nicely for triangular meshes.
- When applied appropriately, differences are difficult to spot visually, though. After three subdivision rounds: Input mesh, Doo-Sabin subdivision, Loop subdivision, Catmull-Clark subdivision, Catmull-Clark subdivision for triangulated faces.



Discussion

- Catmull-Clark subdivision is best for quads, and poor on triangular meshes.
- Loop subdivision and $\sqrt{3}$ subdivision work nicely for triangular meshes.
- When applied appropriately, differences are difficult to spot visually, though. After three subdivision rounds: Input mesh, Doo-Sabin subdivision, Loop subdivision, Catmull-Clark subdivision, Catmull-Clark subdivision for triangulated faces.
- [Stam&Loop (2003), Schaefer&Warren (2005)]: Unified scheme for triangle/quad meshes, with (mostly) C^2 continuity.

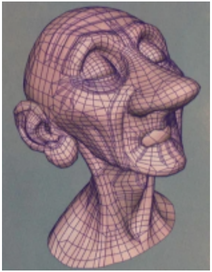


Splines versus Subdivision Surfaces

- NURBS are (still) the number-one contender when it comes to precise non-organic modelling.
- NURBS offer full control over the surface parametrization and its smoothness.

Splines versus Subdivision Surfaces

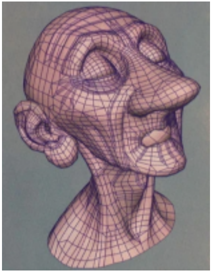
- NURBS are (still) the number-one contender when it comes to precise non-organic modelling.
- NURBS offer full control over the surface parametrization and its smoothness.
- Subdivision surfaces are better suited for describing objects with complex topology because they start with a control mesh of arbitrary (manifold) topology. Spline-based methods (such as NURBS) struggle with complex topology.



[Image credit: “Geri” from Pixar’s “Geri’s Game”]

Splines versus Subdivision Surfaces

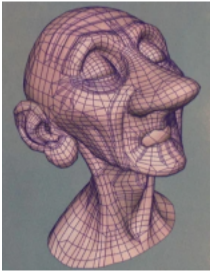
- NURBS are (still) the number-one contender when it comes to precise non-organic modelling.
- NURBS offer full control over the surface parametrization and its smoothness.
- Subdivision surfaces are better suited for describing objects with complex topology because they start with a control mesh of arbitrary (manifold) topology. Spline-based methods (such as NURBS) struggle with complex topology.
- Local refinement to add detail to a localized region of a subdivision surface can be carried out easily by adding faces to appropriate parts of the control mesh.



[Image credit: “Geri” from Pixar’s “Geri’s Game”]

Splines versus Subdivision Surfaces

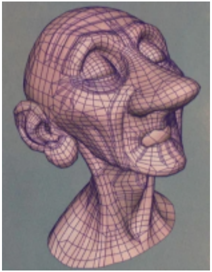
- NURBS are (still) the number-one contender when it comes to precise non-organic modelling.
- NURBS offer full control over the surface parametrization and its smoothness.
- Subdivision surfaces are better suited for describing objects with complex topology because they start with a control mesh of arbitrary (manifold) topology. Spline-based methods (such as NURBS) struggle with complex topology.
- Local refinement to add detail to a localized region of a subdivision surface can be carried out easily by adding faces to appropriate parts of the control mesh.
- Subdivision surfaces are better than polygon meshes because they are smooth and do not look faceted when viewed close up.
- Subdivision surfaces come with level-of-detail modeling.



[Image credit: “Geri” from Pixar’s “Geri’s Game”]

Splines versus Subdivision Surfaces

- NURBS are (still) the number-one contender when it comes to precise non-organic modelling.
- NURBS offer full control over the surface parametrization and its smoothness.
- Subdivision surfaces are better suited for describing objects with complex topology because they start with a control mesh of arbitrary (manifold) topology. Spline-based methods (such as NURBS) struggle with complex topology.
- Local refinement to add detail to a localized region of a subdivision surface can be carried out easily by adding faces to appropriate parts of the control mesh.
- Subdivision surfaces are better than polygon meshes because they are smooth and do not look faceted when viewed close up.



[Image credit: "Geri" from Pixar's "Geri's Game"]

- Subdivision surfaces come with level-of-detail modeling.
- The classic tools and techniques for polygon-mesh modeling can be applied to modeling subdivision control meshes with little extra effort.

Splines versus Subdivision Surfaces

- Since the popular subdivision methods are generalizations of spline-based representations, renderers for subdivision surfaces tend to handle spline surfaces as well.

Splines versus Subdivision Surfaces

- Since the popular subdivision methods are generalizations of spline-based representations, renderers for subdivision surfaces tend to handle spline surfaces as well.
- Subdivision data can be sent to the GPU at coarse resolution and rendered at high resolution.
- Recent GPUs provide hardware support!

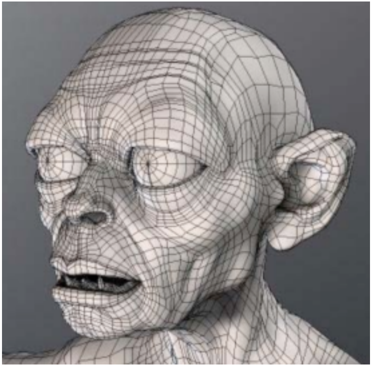
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Splines versus Subdivision Surfaces

- Since the popular subdivision methods are generalizations of spline-based representations, renderers for subdivision surfaces tend to handle spline surfaces as well.
- Subdivision data can be sent to the GPU at coarse resolution and rendered at high resolution.
- Recent GPUs provide hardware support!
- Subdivision meshes tend to be well-suited for finite-element solvers.

Splines versus Subdivision Surfaces

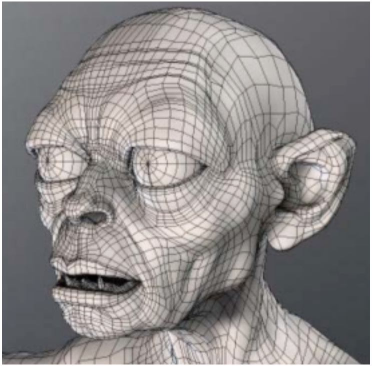
- Since the popular subdivision methods are generalizations of spline-based representations, renderers for subdivision surfaces tend to handle spline surfaces as well.
- Subdivision data can be sent to the GPU at coarse resolution and rendered at high resolution.
- Recent GPUs provide hardware support!
- Subdivision meshes tend to be well-suited for finite-element solvers.
- Subdivision surfaces provide top-quality results for creature modelling in conjunction with bump mapping or displacement mapping.
- In particular, they are not hampered by topological constraints and simplify character animation.



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Splines versus Subdivision Surfaces

- Since the popular subdivision methods are generalizations of spline-based representations, renderers for subdivision surfaces tend to handle spline surfaces as well.
- Subdivision data can be sent to the GPU at coarse resolution and rendered at high resolution.
- Recent GPUs provide hardware support!
- Subdivision meshes tend to be well-suited for finite-element solvers.
- Subdivision surfaces provide top-quality results for creature modelling in conjunction with bump mapping or displacement mapping.
- In particular, they are not hampered by topological constraints and simplify character animation.



- The original model of Gollum (“Lord of the Rings”) was based on NURBS but then converted to subdivision surfaces.

6 Approximation and Interpolation

- Distance Measures
- Interpolation and Approximation of Point Data
- Bernstein Approximation of Functions

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

6 Approximation and Interpolation

- Distance Measures
  - Hausdorff Distance
  - Fréchet Distance
- Interpolation and Approximation of Point Data
- Bernstein Approximation of Functions

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Hausdorff Distance

- Let A, B be two subsets of a metric space X and let $d(p, q)$ denote the distance between two elements $p, q \in X$. E.g., take $\mathbb{R}^n$ and the (standard) Euclidean distance.
- How can we measure how similar A and B are?
- This is a frequently asked question in image processing, solid modeling, computer graphics and computational geometry.

Hausdorff Distance

- Let A, B be two subsets of a metric space X and let $d(p, q)$ denote the distance between two elements $p, q \in X$. E.g., take $\mathbb{R}^n$ and the (standard) Euclidean distance.
- How can we measure how similar A and B are?
- This is a frequently asked question in image processing, solid modeling, computer graphics and computational geometry.
- Note that the classical *minimin* function

$$D(A, B) := \inf_{a \in A} \left(\inf_{b \in B} d(a, b) \right)$$

is a very poor measure of similarity between A and B: One can easily get $D(A, B) = 0$ although A and B need not be similar at all, according to any natural human interpretation of similarity.

Hausdorff Distance

- Let A, B be two subsets of a metric space X and let $d(p, q)$ denote the distance between two elements $p, q \in X$. E.g., take $\mathbb{R}^n$ and the (standard) Euclidean distance.
- How can we measure how similar A and B are?
- This is a frequently asked question in image processing, solid modeling, computer graphics and computational geometry.
- Note that the classical *minimin* function

$$D(A, B) := \inf_{a \in A} \left(\inf_{b \in B} d(a, b) \right)$$

  is a very poor measure of similarity between A and B: One can easily get $D(A, B) = 0$ although A and B need not be similar at all, according to any natural human interpretation of similarity.
- So, can we do any better?

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 185 (Hausdorff distance)

Let A, B be two non-empty subsets of a metric space X and let d be any metric on X. The *directed Hausdorff distance*, $\mathrm{h}(A, B)$, from A to B is defined as

$$\mathrm{h}(A, B) := \sup_{a \in A} \left(\inf_{b \in B} d(a, b) \right).$$

Definition 185 (Hausdorff distance)

Let A, B be two non-empty subsets of a metric space X and let d be any metric on X. The *directed Hausdorff distance*, $\mathrm{h}(A, B)$, from A to B is defined as

$$\mathrm{h}(A, B) := \sup_{a \in A} \left(\inf_{b \in B} d(a, b) \right).$$

The *(symmetric) Hausdorff distance*, $\mathrm{H}(A, B)$, between A and B is defined as

$$\mathrm{H}(A, B) := \max \{\mathrm{h}(A, B), \mathrm{h}(B, A)\}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 185 (Hausdorff distance)

Let A, B be two non-empty subsets of a metric space X and let d be any metric on X. The *directed Hausdorff distance*, $\mathrm{h}(A, B)$, from A to B is defined as

$$\mathrm{h}(A, B) := \sup_{a \in A} \left(\inf_{b \in B} d(a, b) \right).$$

The *(symmetric) Hausdorff distance*, $\mathrm{H}(A, B)$, between A and B is defined as

$$\mathrm{H}(A, B) := \max \{\mathrm{h}(A, B), \mathrm{h}(B, A)\}.$$

- If both A and B are bounded then $\mathrm{H}(A, B)$ is guaranteed to be finite.
- For compact sets we can replace $\inf$ by $\min$ and $\sup$ by $\max$.

Hausdorff Distance

Definition 185 (Hausdorff distance)

Let A, B be two non-empty subsets of a metric space X and let d be any metric on X. The *directed Hausdorff distance*, $\mathrm{h}(A, B)$, from A to B is defined as

$$\mathrm{h}(A, B) := \sup_{a \in A} \left(\inf_{b \in B} d(a, b) \right).$$

The *(symmetric) Hausdorff distance*, $\mathrm{H}(A, B)$, between A and B is defined as

$$\mathrm{H}(A, B) := \max \{\mathrm{h}(A, B), \mathrm{h}(B, A)\}.$$

- If both A and B are bounded then $\mathrm{H}(A, B)$ is guaranteed to be finite.
- For compact sets we can replace $\inf$ by $\min$ and $\sup$ by $\max$.
- The function H defines a metric on the set of all non-empty compact subsets of a metric space X.

Definition 185 (Hausdorff distance)

Let A, B be two non-empty subsets of a metric space X and let d be any metric on X. The *directed Hausdorff distance*, $\mathrm{h}(A, B)$, from A to B is defined as

$$\mathrm{h}(A, B) := \sup_{a \in A} \left(\inf_{b \in B} d(a, b) \right).$$

The *(symmetric) Hausdorff distance*, $\mathrm{H}(A, B)$, between A and B is defined as

$$\mathrm{H}(A, B) := \max \left\{ \mathrm{h}(A, B), \mathrm{h}(B, A) \right\}.$$

- If both A and B are bounded then $\mathrm{H}(A, B)$ is guaranteed to be finite.
- For compact sets we can replace $\inf$ by $\min$ and $\sup$ by $\max$.
- The function H defines a metric on the set of all non-empty compact subsets of a metric space X.
- For sets of n points in $\mathbb{R}^2$, the Hausdorff distance can be computed in time $O(n \log n)$, using a Voronoi-based approach $\longrightarrow$ computational geometry.
- A common variation is the *Hausdorff distance under translation*.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Hausdorff Distance

Definition 185 (Hausdorff distance)

Let A, B be two non-empty subsets of a metric space X and let d be any metric on X. The *directed Hausdorff distance*, $\mathrm{h}(A, B)$, from A to B is defined as

$$\mathrm{h}(A, B) := \sup_{a \in A} \left(\inf_{b \in B} d(a, b) \right).$$

The *(symmetric) Hausdorff distance*, $\mathrm{H}(A, B)$, between A and B is defined as

$$\mathrm{H}(A, B) := \max \{\mathrm{h}(A, B), \mathrm{h}(B, A)\}.$$

- If both A and B are bounded then $\mathrm{H}(A, B)$ is guaranteed to be finite.
- For compact sets we can replace $\inf$ by $\min$ and $\sup$ by $\max$.
- The function H defines a metric on the set of all non-empty compact subsets of a metric space X.
- For sets of n points in $\mathbb{R}^2$, the Hausdorff distance can be computed in time $O(n \log n)$, using a Voronoi-based approach $\longrightarrow$ computational geometry.
- A common variation is the *Hausdorff distance under translation*.
- The Hausdorff distance does not capture any form of orientation or continuity, as we might be interested in when matching curves or surfaces.

Definition 186 (Fréchet distance)

Consider a closed interval $I \subset \mathbb{R}$ and two curves $\beta, \gamma \colon I \to \mathbb{R}^n$. The *Fréchet distance* between $\beta(I)$ and $\gamma(I)$ is defined as

$$\mathrm{Fr}(\beta, \gamma) := \inf_{\sigma, \tau} \max_{t \in I} \|\beta(\sigma(t)) - \gamma(\tau(t))\|,$$

where $\sigma, \tau \colon I \to I$ range over all continuous and monotonously increasing functions that map I to I such that $\sigma(I) = I$ and $\tau(I) = I$.

Definition 186 (Fréchet distance)

Consider a closed interval $I \subset \mathbb{R}$ and two curves $\beta, \gamma\colon I \to \mathbb{R}^n$. The *Fréchet distance* between $\beta(I)$ and $\gamma(I)$ is defined as

$$\operatorname{Fr}(\beta, \gamma) := \inf_{\sigma,\tau} \max_{t \in I} \|\beta(\sigma(t)) - \gamma(\tau(t))\|,$$

where $\sigma, \tau\colon I \to I$ range over all continuous and monotonously increasing functions that map I to I such that $\sigma(I) = I$ and $\tau(I) = I$.

- Popular interpretation [Alt&Godau 1995]: Suppose that a person is walking a dog. Assume the person is walking on one curve and the dog on another curve. Both can adjust their speeds but are not allowed to move backwards.

Definition 186 (Fréchet distance)

Consider a closed interval $I \subset \mathbb{R}$ and two curves $\beta, \gamma\colon I \to \mathbb{R}^n$. The *Fréchet distance* between $\beta(I)$ and $\gamma(I)$ is defined as

$$\mathrm{Fr}(\beta, \gamma) := \inf_{\sigma,\tau} \max_{t \in I} \|\beta(\sigma(t)) - \gamma(\tau(t))\|,$$

where $\sigma, \tau\colon I \to I$ range over all continuous and monotonously increasing functions that map I to I such that $\sigma(I) = I$ and $\tau(I) = I$.

- Popular interpretation [Alt&Godau 1995]: Suppose that a person is walking a dog. Assume the person is walking on one curve and the dog on another curve. Both can adjust their speeds but are not allowed to move backwards.
- We can think of the parameter t as time: Then $\beta(\sigma(t))$ is the position of the person and $\gamma(\tau(t))$ is the position of the dog at time t. The length of the leash between them at time t is the distance between $\beta(\sigma(t))$ and $\gamma(\tau(t))$.

Fréchet Distance

Definition 186 (Fréchet distance)

Consider a closed interval $I \subset \mathbb{R}$ and two curves $\beta, \gamma \colon I \to \mathbb{R}^n$. The *Fréchet distance* between $\beta(I)$ and $\gamma(I)$ is defined as

$$\mathrm{Fr}(\beta, \gamma) := \inf_{\sigma,\tau} \max_{t \in I} \|\beta(\sigma(t)) - \gamma(\tau(t))\|,$$

where $\sigma, \tau \colon I \to I$ range over all continuous and monotonously increasing functions that map I to I such that $\sigma(I) = I$ and $\tau(I) = I$.

- Popular interpretation [Alt&Godau 1995]: Suppose that a person is walking a dog. Assume the person is walking on one curve and the dog on another curve. Both can adjust their speeds but are not allowed to move backwards.
- We can think of the parameter t as time: Then $\beta(\sigma(t))$ is the position of the person and $\gamma(\tau(t))$ is the position of the dog at time t. The length of the leash between them at time t is the distance between $\beta(\sigma(t))$ and $\gamma(\tau(t))$.
- Then the Fréchet distance of the two curves is the minimum leash length necessary to keep the person and the dog connected at all times $t \in I$.

Fréchet Distance

Definition 186 (Fréchet distance)

Consider a closed interval $I \subset \mathbb{R}$ and two curves $\beta, \gamma \colon I \to \mathbb{R}^n$. The *Fréchet distance* between $\beta(I)$ and $\gamma(I)$ is defined as

$$\mathrm{Fr}(\beta, \gamma) := \inf_{\sigma,\tau} \max_{t \in I} \|\beta(\sigma(t)) - \gamma(\tau(t))\|,$$

where $\sigma, \tau \colon I \to I$ range over all continuous and monotonously increasing functions that map I to I such that $\sigma(I) = I$ and $\tau(I) = I$.

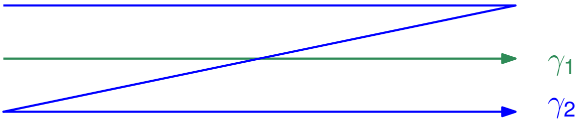
- Popular interpretation [Alt&Godau 1995]: Suppose that a person is walking a dog. Assume the person is walking on one curve and the dog on another curve. Both can adjust their speeds but are not allowed to move backwards.
- We can think of the parameter t as time: Then $\beta(\sigma(t))$ is the position of the person and $\gamma(\tau(t))$ is the position of the dog at time t. The length of the leash between them at time t is the distance between $\beta(\sigma(t))$ and $\gamma(\tau(t))$.
- Then the Fréchet distance of the two curves is the minimum leash length necessary to keep the person and the dog connected at all times $t \in I$.
- Note that we do not demand strict monotonicity for either σ or τ.

Fréchet Distance

- The Fréchet distance between two curves may be arbitrarily larger than the Hausdorff distance between them.

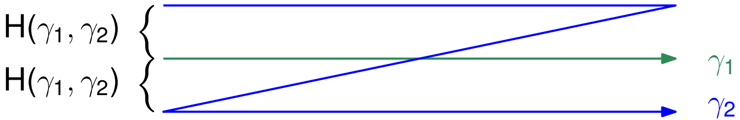
Fréchet Distance

- The Fréchet distance between two curves may be arbitrarily larger than the Hausdorff distance between them.

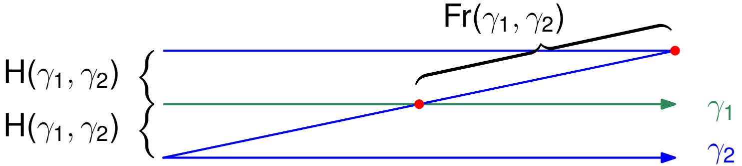


Fréchet Distance

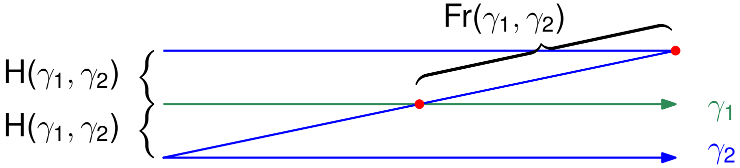
- The Fréchet distance between two curves may be arbitrarily larger than the Hausdorff distance between them.



- The Fréchet distance between two curves may be arbitrarily larger than the Hausdorff distance between them.

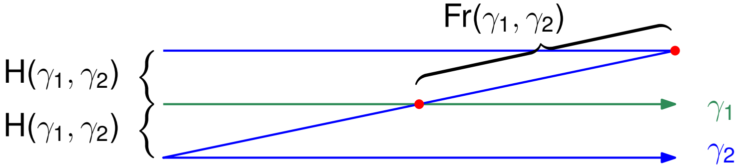


- The Fréchet distance between two curves may be arbitrarily larger than the Hausdorff distance between them.
- [Alt&Godau 1995] give a (complicated) algorithm that computes the exact Fréchet distance between two polygonal curves in time $O(nm \log(nm))$, where n and m are the number of vertices of the polygonal curves.



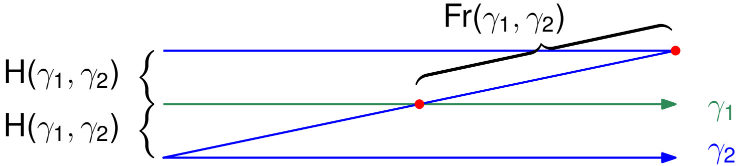
Fréchet Distance

- The Fréchet distance between two curves may be arbitrarily larger than the Hausdorff distance between them.
- [Alt&Godau 1995] give a (complicated) algorithm that computes the exact Fréchet distance between two polygonal curves in time $O(nm \log(nm))$, where n and m are the number of vertices of the polygonal curves.
- [Bringmann 2014] shows that, conditional on the Strong Exponential Time Hypothesis (SETH), there cannot exist an $O(n^{2-\varepsilon})$ algorithm for deciding whether two n-vertex polygonal curves have a Fréchet distance at most δ. However, in practice a fast computation can be engineered [Bringmann et al. 2021].



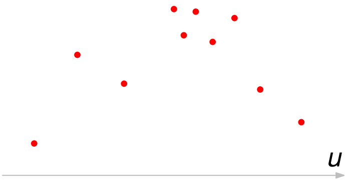
Fréchet Distance

- The Fréchet distance between two curves may be arbitrarily larger than the Hausdorff distance between them.
- [Alt&Godau 1995] give a (complicated) algorithm that computes the exact Fréchet distance between two polygonal curves in time $O(nm\log(nm))$, where n and m are the number of vertices of the polygonal curves.
- [Bringmann 2014] shows that, conditional on the Strong Exponential Time Hypothesis (SETH), there cannot exist an $O(n^{2-\varepsilon})$ algorithm for deciding whether two n-vertex polygonal curves have a Fréchet distance at most δ. However, in practice a fast computation can be engineered [Bringmann et al. 2021].
- The same problem is $\mathcal{NP}$-hard for triangulated surfaces. Only a variant, the so-called *weak Fréchet distance*, can be computed in polynomial time [Alt&Buchin 2010].



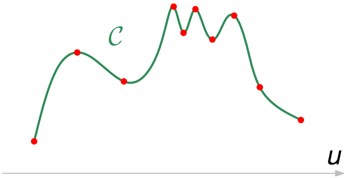
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- For $m \in \mathbb{N}_0$, we are given $m + 1$ points $q_0, q_1, \ldots, q_m \in \mathbb{R}^n$, possibly with matching parameter values $u_0 < u_1 < \ldots < u_m$.



Interpolation Versus Approximation

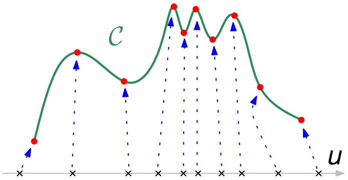
- For $m \in \mathbb{N}_0$, we are given $m+1$ points $q_0, q_1, \ldots, q_m \in \mathbb{R}^n$, possibly with matching parameter values $u_0 < u_1 < \ldots < u_m$.
- For an interpolation of $q_0, q_1, \ldots, q_m$ we seek a curve $\mathcal{C}$ such that either
  - $\mathcal{C}(x_i) = q_i$ for arbitrary $x_i \in \mathbb{R}$, for all $i \in \{0, 1, \ldots, m\}$,



UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Interpolation Versus Approximation

- For $m \in \mathbb{N}_0$, we are given $m+1$ points $q_0, q_1, \ldots, q_m \in \mathbb{R}^n$, possibly with matching parameter values $u_0 < u_1 < \ldots < u_m$.
- For an interpolation of $q_0, q_1, \ldots, q_m$ we seek a curve $\mathcal{C}$ such that either
  - $\mathcal{C}(x_i) = q_i$ for arbitrary $x_i \in \mathbb{R}$, for all $i \in \{0, 1, \ldots, m\}$, or
  - $\mathcal{C}(u_i) = q_i$ for all $i \in \{0, 1, \ldots, m\}$.

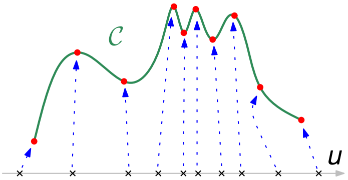


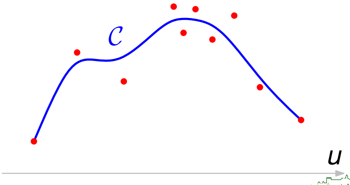
Interpolation Versus Approximation

- For $m \in \mathbb{N}_0$, we are given $m+1$ points $q_0, q_1, \ldots, q_m \in \mathbb{R}^n$, possibly with matching parameter values $u_0 < u_1 < \ldots < u_m$.

- For an interpolation of $q_0, q_1, \ldots, q_m$ we seek a curve $\mathcal{C}$ such that either
  - $\mathcal{C}(x_i) = q_i$ for arbitrary $x_i \in \mathbb{R}$, for all $i \in \{0, 1, \ldots, m\}$, or
  - $\mathcal{C}(u_i) = q_i$ for all $i \in \{0, 1, \ldots, m\}$.

- For an approximation of $q_0, q_1, \ldots, q_m$ we seek a curve $\mathcal{C}$ such that the distance between $\mathcal{C}$ and $q_0, q_1, \ldots, q_m$ is smaller than a user-specified threshold relative to some distance measure.

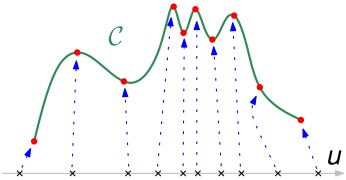


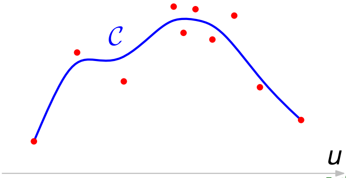


UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Interpolation Versus Approximation

- For $m \in \mathbb{N}_0$, we are given $m+1$ points $q_0, q_1, \ldots, q_m \in \mathbb{R}^n$, possibly with matching parameter values $u_0 < u_1 < \ldots < u_m$.
- For an interpolation of $q_0, q_1, \ldots, q_m$ we seek a curve $\mathcal{C}$ such that either
  - $\mathcal{C}(x_i) = q_i$ for arbitrary $x_i \in \mathbb{R}$, for all $i \in \{0, 1, \ldots, m\}$, or
  - $\mathcal{C}(u_i) = q_i$ for all $i \in \{0, 1, \ldots, m\}$.
- For an approximation of $q_0, q_1, \ldots, q_m$ we seek a curve $\mathcal{C}$ such that the distance between $\mathcal{C}$ and $q_0, q_1, \ldots, q_m$ is smaller than a user-specified threshold relative to some distance measure.
- Similarly for approximation/interpolation by a surface rather than a curve.

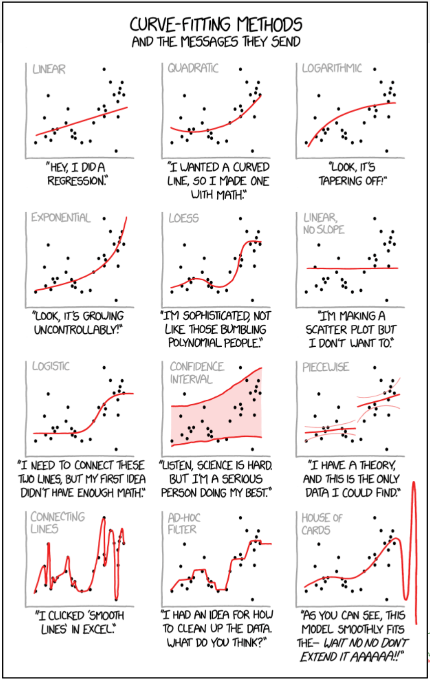




UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Humorous View of Approximation

[Image credit: https://xkcd.com]



Definition 187 (Lagrange polynomial)

For $m \in \mathbb{N}$, consider $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and let $i \in \{0, 1, \ldots, m\}$. Then the i-th *Lagrange polynomial* of degree m is defined as

$$L_{i,m}(u) := \prod_{j=0, i \neq j}^{m} \frac{u - u_j}{u_i - u_j}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 187 (Lagrange polynomial)

For $m \in \mathbb{N}$, consider $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and let $i \in \{0, 1, \ldots, m\}$. Then the i-th *Lagrange polynomial* of degree m is defined as

$$L_{i,m}(u) := \prod_{j=0, i \neq j}^{m} \frac{u - u_j}{u_i - u_j}.$$

- That is,

$$L_{i,m}(u) = \frac{u - u_0}{u_i - u_0} \cdot \frac{u - u_1}{u_i - u_1} \cdot \ldots \cdot \frac{u - u_{i-1}}{u_i - u_{i-1}} \cdot \frac{u - u_{i+1}}{u_i - u_{i+1}} \cdot \ldots \cdot \frac{u - u_{m-1}}{u_i - u_{m-1}} \cdot \frac{u - u_m}{u_i - u_m}.$$

Definition 187 (Lagrange polynomial)

For $m \in \mathbb{N}$, consider $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and let $i \in \{0, 1, \ldots, m\}$. Then the i-th *Lagrange polynomial* of degree m is defined as

$$L_{i,m}(u) := \prod_{j=0, i \neq j}^{m} \frac{u - u_j}{u_i - u_j}.$$

- That is,

$$L_{i,m}(u) = \frac{u - u_0}{u_i - u_0} \cdot \frac{u - u_1}{u_i - u_1} \cdot \ldots \cdot \frac{u - u_{i-1}}{u_i - u_{i-1}} \cdot \frac{u - u_{i+1}}{u_i - u_{i+1}} \cdot \ldots \cdot \frac{u - u_{m-1}}{u_i - u_{m-1}} \cdot \frac{u - u_m}{u_i - u_m}.$$

Definition 188 (Lagrange interpolation)

For $m \in \mathbb{N}$, consider $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$. Then the *Lagrange interpolation* of $q_0, q_1, \ldots, q_m$ is given by

$$\mathcal{L}(u) := \sum_{i=0}^{m} L_{i,m}(u) q_i.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 189

For $m \in \mathbb{N}$, let $\mathcal{L}$ be defined for $m + 1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m + 1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 188. Then $\mathcal{L}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Lemma 189

For $m \in \mathbb{N}$, let $\mathcal{L}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 188. Then $\mathcal{L}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have

$$L_{i,m}(u_k) = \prod_{j=0, i\neq j}^{m} \frac{u_k - u_j}{u_i - u_j} = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k, \\ 1 & \text{if } i = k. \end{cases}$$

Lemma 189

For $m \in \mathbb{N}$, let $\mathcal{L}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 188. Then $\mathcal{L}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have

$$L_{i,m}(u_k) = \prod_{j=0, i \neq j}^{m} \frac{u_k - u_j}{u_i - u_j} = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k, \\ 1 & \text{if } i = k. \end{cases}$$

Hence,

$$\mathcal{L}(u_k) = \sum_{i=0}^{m} L_{i,m}(u_k) q_i = \sum_{i=0}^{m} \delta_{ik} q_i = q_k.$$

□

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lagrange Interpolation

Lemma 189

For $m \in \mathbb{N}$, let $\mathcal{L}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 188. Then $\mathcal{L}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof : For all $k \in \{0, 1, \ldots, m\}$, we have

$$L_{i,m}(u_k) = \prod_{j=0, i \neq j}^{m} \frac{u_k - u_j}{u_i - u_j} = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k, \\ 1 & \text{if } i = k. \end{cases}$$

Hence,

$$\mathcal{L}(u_k) = \sum_{i=0}^{m} L_{i,m}(u_k) q_i = \sum_{i=0}^{m} \delta_{ik} q_i = q_k.$$

□

Corollary 190

For $m \in \mathbb{N}$, the Lagrange polynomials $L_{0,m}, L_{1,m}, \ldots, L_{m,m}$ form a basis of the vector space of all polynomials of degree at most m.

Lemma 189

For $m \in \mathbb{N}$, let $\mathcal{L}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 188. Then $\mathcal{L}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have

$$L_{i,m}(u_k) = \prod_{j=0, i\neq j}^{m} \frac{u_k - u_j}{u_i - u_j} = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k, \\ 1 & \text{if } i = k. \end{cases}$$

Hence,

$$\mathcal{L}(u_k) = \sum_{i=0}^{m} L_{i,m}(u_k) q_i = \sum_{i=0}^{m} \delta_{ik} q_i = q_k.$$

□

Corollary 190

For $m \in \mathbb{N}$, the Lagrange polynomials $L_{0,m}, L_{1,m}, \ldots, L_{m,m}$ form a basis of the vector space of all polynomials of degree at most m.

Sketch of proof: Exactly one polynomial of degree m interpolates $m+1$ data points!

Definition 191 (Newton polynomial)

For $m \in \mathbb{N}$, consider $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and let $i \in \{0, 1, \ldots, m\}$. Then the i-th *Newton polynomial* is defined as

$$l_i(u) := \prod_{j=0}^{i-1} (u - u_j) \qquad \text{with, by convention, } l_0(u) := 1.$$

Newton Interpolation

Definition 191 (Newton polynomial)

For $m \in \mathbb{N}$, consider $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and let $i \in \{0, 1, \ldots, m\}$. Then the i-th *Newton polynomial* is defined as

$$l_i(u) := \prod_{j=0}^{i-1} (u - u_j) \qquad \text{with, by convention, } l_0(u) := 1.$$

Definition 192 (Newton interpolation)

For $m \in \mathbb{N}$, consider $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$. Then the *Newton interpolation* of $q_0, q_1, \ldots, q_m$ is given by

$$\mathcal{I}(u) := \sum_{i=0}^{m} l_i(u) p_i,$$

with

$$p_i := \begin{cases} q_i & \text{for } i = 0, \\ \frac{q_i - \sum_{j=0}^{i-1} l_j(u_i) p_j}{l_i(u_i)} & \text{for } i > 0. \end{cases}$$

Lemma 193

For $m \in \mathbb{N}$, let $\mathcal{I}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 192. Then $\mathcal{I}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Lemma 193

For $m \in \mathbb{N}$, let $\mathcal{I}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 192. Then $\mathcal{I}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof : For all $k \in \{0, 1, \ldots, m\}$, we have for all $i > 1$

$$l_i(u_k) = \prod_{j=0}^{i-1}(u_k - u_j) \quad \begin{cases} = 0 & \text{if } k \le i-1, \\ \neq 0 & \text{if } k \ge i. \end{cases}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 193

For $m \in \mathbb{N}$, let $\mathcal{I}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 192. Then $\mathcal{I}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have for all $i > 1$

$$l_i(u_k) = \prod_{j=0}^{i-1}(u_k - u_j) \quad \begin{cases} = 0 & \text{if } k \leq i-1, \\ \neq 0 & \text{if } k \geq i. \end{cases}$$

We have

$$\mathcal{I}(u_0) = 1 \cdot p_0 = q_0,$$

Lemma 193

For $m \in \mathbb{N}$, let $\mathcal{I}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 192. Then $\mathcal{I}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have for all $i > 1$

$$l_i(u_k) = \prod_{j=0}^{i-1}(u_k - u_j) \quad \begin{cases} = 0 & \text{if } k \leq i-1, \\ \neq 0 & \text{if } k \geq i. \end{cases}$$

We have

$$\mathcal{I}(u_0) = 1 \cdot p_0 = q_0,$$

and for each $1 \leq k \leq m$

$$\mathcal{I}(u_k) = \sum_{i=0}^{m} l_i(u_k)p_i = \sum_{i=0}^{k} l_i(u_k)p_i$$

Lemma 193

For $m \in \mathbb{N}$, let $\mathcal{I}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 192. Then $\mathcal{I}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have for all $i > 1$

$$l_i(u_k) = \prod_{j=0}^{i-1}(u_k - u_j) \quad \begin{cases} = 0 & \text{if } k \leq i-1, \\ \neq 0 & \text{if } k \geq i. \end{cases}$$

We have

$$\mathcal{I}(u_0) = 1 \cdot p_0 = q_0,$$

and for each $1 \leq k \leq m$

$$\mathcal{I}(u_k) = \sum_{i=0}^{m} l_i(u_k)p_i = \sum_{i=0}^{k} l_i(u_k)p_i = \sum_{i=0}^{k-1} l_i(u_k)p_i + l_k(u_k)p_k$$

Lemma 193

For $m \in \mathbb{N}$, let $\mathcal{I}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 192. Then $\mathcal{I}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have for all $i > 1$

$$l_i(u_k) = \prod_{j=0}^{i-1}(u_k - u_j) \quad \begin{cases} = 0 & \text{if } k \leq i-1, \\ \neq 0 & \text{if } k \geq i. \end{cases}$$

We have

$$\mathcal{I}(u_0) = 1 \cdot p_0 = q_0,$$

and for each $1 \leq k \leq m$

$$\mathcal{I}(u_k) = \sum_{i=0}^{m} l_i(u_k)p_i = \sum_{i=0}^{k} l_i(u_k)p_i = \sum_{i=0}^{k-1} l_i(u_k)p_i + l_k(u_k)p_k$$
$$= \sum_{i=0}^{k-1} l_i(u_k)p_i + l_k(u_k) \cdot \frac{q_k - \sum_{j=0}^{k-1} l_j(u_k)p_j}{l_k(u_k)}$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Lemma 193

For $m \in \mathbb{N}$, let $\mathcal{I}$ be defined for $m+1$ parameter values $u_0 < u_1 < \ldots < u_m$ and $m+1$ data points $q_0, q_1, \ldots, q_m$, as given in Def. 192. Then $\mathcal{I}(u_k) = q_k$ for all $k \in \{0, 1, \ldots, m\}$.

Proof: For all $k \in \{0, 1, \ldots, m\}$, we have for all $i > 1$

$$l_i(u_k) = \prod_{j=0}^{i-1}(u_k - u_j) \quad \begin{cases} = 0 & \text{if } k \leq i-1, \\ \neq 0 & \text{if } k \geq i. \end{cases}$$

We have

$$\mathcal{I}(u_0) = 1 \cdot p_0 = q_0,$$

and for each $1 \leq k \leq m$

$$\mathcal{I}(u_k) = \sum_{i=0}^{m} l_i(u_k)p_i = \sum_{i=0}^{k} l_i(u_k)p_i = \sum_{i=0}^{k-1} l_i(u_k)p_i + l_k(u_k)p_k$$

$$= \sum_{i=0}^{k-1} l_i(u_k)p_i + l_k(u_k) \cdot \frac{q_k - \sum_{j=0}^{k-1} l_j(u_k)p_j}{l_k(u_k)} = q_k.$$

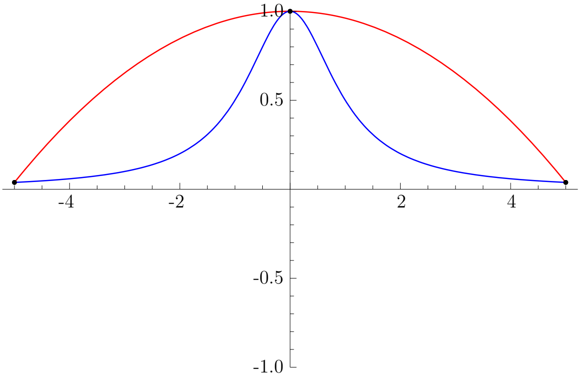
Limits of Lagrange Interpolation and Newton Interpolation

- Sampling of a function f and subsequent Lagrange interpolation may yield an extremely poor approximation of f even if f is continuously differentiable.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

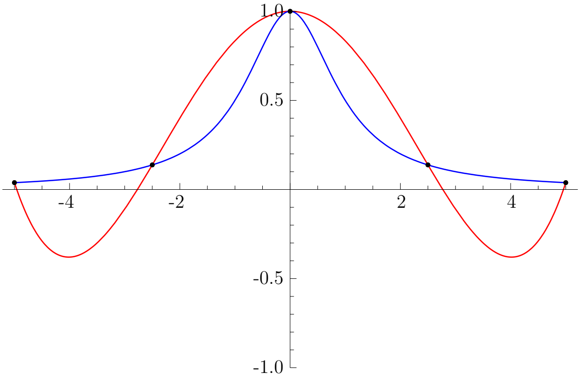
Limits of Lagrange Interpolation and Newton Interpolation

- Sampling of a function f and subsequent Lagrange interpolation may yield an extremely poor approximation of f even if f is continuously differentiable.
- C. Runge: Consider $f(x) := \frac{1}{1+x^2}$ and $n+1$ uniform samples within $[-5, 5]$, with $n := 2$.



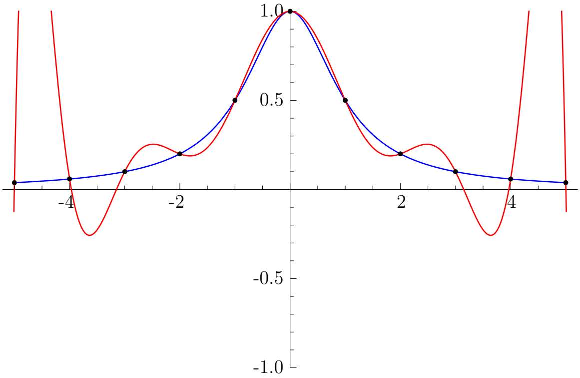
Limits of Lagrange Interpolation and Newton Interpolation

- Sampling of a function f and subsequent Lagrange interpolation may yield an extremely poor approximation of f even if f is continuously differentiable.
- C. Runge: Consider $f(x) := \frac{1}{1+x^2}$ and $n+1$ uniform samples within $[-5, 5]$, with $n := 4$.



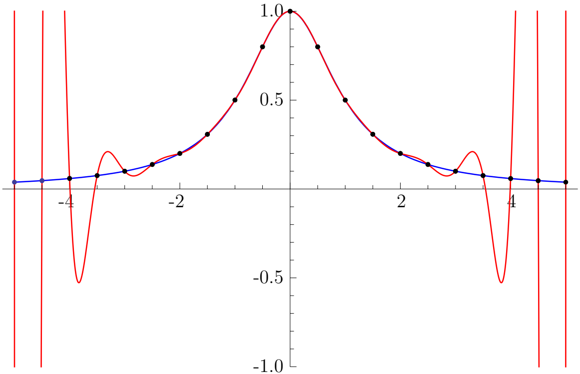
Limits of Lagrange Interpolation and Newton Interpolation

- Sampling of a function f and subsequent Lagrange interpolation may yield an extremely poor approximation of f even if f is continuously differentiable.
- C. Runge: Consider $f(x) := \frac{1}{1+x^2}$ and $n+1$ uniform samples within $[-5, 5]$, with $n := 10$.



Limits of Lagrange Interpolation and Newton Interpolation

- Sampling of a function f and subsequent Lagrange interpolation may yield an extremely poor approximation of f even if f is continuously differentiable.
- C. Runge: Consider $f(x) := \frac{1}{1+x^2}$ and $n+1$ uniform samples within $[-5, 5]$, with $n := 20$.
- Similar problems occur for Newton interpolation.



B-Spline Interpolation

- Let $k \in \mathbb{N}_0$ and suppose that we are looking for $n+1$ control points $p_0, p_1, \ldots, p_n$ and a knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ such that the B-spline curve $\mathcal{B}$ of degree k defined by $p_0, p_1, \ldots, p_n$ and τ interpolates $q_0, q_1, \ldots, q_m$, with $\mathcal{B}(u_i) = q_i$ for all $i \in \{0, 1, \ldots, m\}$ and some given $u_0 < u_1 < \ldots < u_m$.

B-Spline Interpolation

- Let $k \in \mathbb{N}_0$ and suppose that we are looking for $n+1$ control points $p_0, p_1, \ldots, p_n$ and a knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ such that the B-spline curve $\mathcal{B}$ of degree k defined by $p_0, p_1, \ldots, p_n$ and τ interpolates $q_0, q_1, \ldots, q_m$, with $\mathcal{B}(u_i) = q_i$ for all $i \in \{0, 1, \ldots, m\}$ and some given $u_0 < u_1 < \ldots < u_m$.
- If $n = m$, then we get the following system of equations:

$$\underbrace{\begin{pmatrix} N_{0,k}(u_0) & \cdots & N_{n,k}(u_0) \\ \vdots & & \vdots \\ N_{0,k}(u_n) & \cdots & N_{n,k}(u_n) \end{pmatrix}}_{=:\mathbf{N}} \cdot \underbrace{\begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}}_{=:p} = \underbrace{\begin{pmatrix} q_0 \\ \vdots \\ q_n \end{pmatrix}}_{=:q}$$

- This interpolation problem can be solved if the *collocation matrix* **N** is invertible.

- Let $k \in \mathbb{N}_0$ and suppose that we are looking for $n+1$ control points $p_0, p_1, \ldots, p_n$ and a knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ such that the B-spline curve $\mathcal{B}$ of degree k defined by $p_0, p_1, \ldots, p_n$ and τ interpolates $q_0, q_1, \ldots, q_m$, with $\mathcal{B}(u_i) = q_i$ for all $i \in \{0, 1, \ldots, m\}$ and some given $u_0 < u_1 < \ldots < u_m$.
- If $n = m$, then we get the following system of equations:

$$\underbrace{\begin{pmatrix} N_{0,k}(u_0) & \cdots & N_{n,k}(u_0) \\ \vdots & & \vdots \\ N_{0,k}(u_n) & \cdots & N_{n,k}(u_n) \end{pmatrix}}_{=:\mathbf{N}} \cdot \underbrace{\begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}}_{=:p} = \underbrace{\begin{pmatrix} q_0 \\ \vdots \\ q_n \end{pmatrix}}_{=:q}$$

- This interpolation problem can be solved if the *collocation matrix* **N** is invertible.

Lemma 194 (Schönberg-Whitney)

The collocation matrix **N** is invertible if and only if if all its diagonal elements $N_{i,k}(u_i)$ are non-zero.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Let $k \in \mathbb{N}_0$ and suppose that we are looking for $n+1$ control points $p_0, p_1, \ldots, p_n$ and a knot vector $\tau := (t_0, t_1, \ldots, t_{n+k+1})$ such that the B-spline curve $\mathcal{B}$ of degree k defined by $p_0, p_1, \ldots, p_n$ and τ interpolates $q_0, q_1, \ldots, q_m$, with $\mathcal{B}(u_i) = q_i$ for all $i \in \{0, 1, \ldots, m\}$ and some given $u_0 < u_1 < \ldots < u_m$.
- If $n = m$, then we get the following system of equations:

$$\underbrace{\begin{pmatrix} N_{0,k}(u_0) & \cdots & N_{n,k}(u_0) \\ \vdots & & \vdots \\ N_{0,k}(u_n) & \cdots & N_{n,k}(u_n) \end{pmatrix}}_{=:\mathbf{N}} \cdot \underbrace{\begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}}_{=:p} = \underbrace{\begin{pmatrix} q_0 \\ \vdots \\ q_n \end{pmatrix}}_{=:q}$$

- This interpolation problem can be solved if the *collocation matrix* $\mathbf{N}$ is invertible.

Lemma 194 (Schönberg-Whitney)

The collocation matrix $\mathbf{N}$ is invertible if and only if if all its diagonal elements $N_{i,k}(u_i)$ are non-zero.

- If the multiplicity of all knots is at most k then Lemma 129 implies the condition $t_i < u_i < t_{i+k+1}$ and that $\mathbf{N}$ is a sparse band matrix without negative elements.
- Fast and numerically reliable algorithms exist for computing the inverse of $\mathbf{N}$.

- Most applications do not require specific parameter values u_i.
- In such a case, one can fix the knots t_i, and choose u_i as follows ("Greville-abscissae"):

$$u_i := \frac{1}{k} \sum_{j=1}^{k} t_{i+j} \quad \text{for all } i \in \{0, 1, \ldots, n\}.$$

- Note that t_i and t_{i+k+1} do not enter the definition of u_i.

B-Spline Interpolation

- Most applications do not require specific parameter values u_i.
- In such a case, one can fix the knots t_i, and choose u_i as follows ("Greville-abscissae"):

$$u_i := \frac{1}{k} \sum_{j=1}^{k} t_{i+j} \quad \text{for all } i \in \{0, 1, \ldots, n\}.$$

- Note that t_i and t_{i+k+1} do not enter the definition of u_i.
- Of course,

$$t_i \leq t_{i+1} \leq \frac{1}{k}(t_{i+1} + \cdots + t_{i+k}) \leq t_{i+k} \leq t_{i+k+1},$$

thus meeting the Schönberg-Whitney condition of Lem. 194. Equality would only occur if an inner knot has multiplicity $k+1$. (But then the B-spline would be discontinuous!)

Effects of Parameters and Knots

- Since a B-spline has continuous speed and acceleration (for $k \geq 3$), it is obvious that the parameter values u_i should bear a meaningful relation to the distances between the data points. Otherwise, overshooting is bound to occur!

Effects of Parameters and Knots

- Since a B-spline has continuous speed and acceleration (for $k \geq 3$), it is obvious that the parameter values u_i should bear a meaningful relation to the distances between the data points. Otherwise, overshooting is bound to occur!
- Consider

$$u_0 := 0 \quad \text{and } u_{i+1} := u_i + \Delta_i \text{ for all } i \in \{1, \ldots, m-1\},$$

with

$$\Delta_i := \|q_i - q_{i-1}\|^p \text{ for some } p \in [0, 1] \text{ and all } i \in \{1, \ldots, m-1\}.$$

- These parameter values are known as *uniform* if $p = 0$, *centripetal* if $p = \frac{1}{2}$, and *chordal* if $p = 1$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Effects of Parameters and Knots

- Since a B-spline has continuous speed and acceleration (for $k \geq 3$), it is obvious that the parameter values u_i should bear a meaningful relation to the distances between the data points. Otherwise, overshooting is bound to occur!
- Consider

$$u_0 := 0 \quad \text{and } u_{i+1} := u_i + \Delta_i \ \text{ for all } i \in \{1, \ldots, m-1\},$$

with

$$\Delta_i := \|q_i - q_{i-1}\|^p \ \text{ for some } p \in [0, 1] \text{ and all } i \in \{1, \ldots, m-1\}.$$

- These parameter values are known as *uniform* if $p = 0$, *centripetal* if $p = \frac{1}{2}$, and *chordal* if $p = 1$.
- Suitable knots that meet the Schönberg-Whitney conditions (Lem. 194) are defined as follows:

$$t_i := \frac{1}{k}(u_{i-k} + u_{i-k+1} + \ldots + u_{i-1})$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

B-Spline Approximation

- If $m > n$, i.e., if there are more data points than control points, then the linear system $\mathbf{N}p = q$ is over-determined and a solution need not exist.

B-Spline Approximation

- If $m > n$, i.e., if there are more data points than control points, then the linear system $\mathbf{N}p = q$ is over-determined and a solution need not exist.
- One popular option is a least-squares fit, which is achieved if

$$\mathbf{N}^T\mathbf{N}p = \mathbf{N}^T q.$$

- Hence, if $\mathbf{N}^T\mathbf{N}$ is invertible then we get

$$p = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T q.$$

B-Spline Approximation

- If $m > n$, i.e., if there are more data points than control points, then the linear system $\mathbf{N}p = q$ is over-determined and a solution need not exist.
- One popular option is a least-squares fit, which is achieved if

$$\mathbf{N}^T\mathbf{N}p = \mathbf{N}^T q.$$

- Hence, if $\mathbf{N}^T\mathbf{N}$ is invertible then we get

$$p = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T q.$$

- An extension of the Schönberg-Whitney Lem. 194 tells us that the matrix $\mathbf{N}^T\mathbf{N}$ is invertible exactly if the Schönberg-Whitney conditions are met:

Lemma 195

The matrix $\mathbf{N}^T\mathbf{N}$ is invertible if and only if $t_i \leq u_i < t_{i+k+1}$, for all $i \in \{0, 1, \ldots, n\}$.

6 Approximation and Interpolation

- Distance Measures
- Interpolation and Approximation of Point Data
- Bernstein Approximation of Functions

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 196 (Bernstein polynomial)

For $n \in \mathbb{N}_0$, a *Bernstein polynomial* of degree n is a linear combination of Bernstein basis polynomials of degree n:

$$B_n(x) := \sum_{i=0}^{n} \mu_i B_{i,n}(x), \quad \text{with } \mu_0, \mu_1, \ldots, \mu_n \in \mathbb{R}.$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Polynomials

Definition 196 (Bernstein polynomial)

For $n \in \mathbb{N}_0$, a *Bernstein polynomial* of degree n is a linear combination of Bernstein basis polynomials of degree n:

$$B_n(x) := \sum_{i=0}^{n} \mu_i B_{i,n}(x), \quad \text{with } \mu_0, \mu_1, \ldots, \mu_n \in \mathbb{R}.$$

- Hence, every polynomial (in power basis) can be seen as a Bernstein polynomial, albeit with unknown scalars for the linear combination.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Bernstein Polynomials

Definition 196 (Bernstein polynomial)

For $n \in \mathbb{N}_0$, a *Bernstein polynomial* of degree n is a linear combination of Bernstein basis polynomials of degree n:

$$B_n(x) := \sum_{i=0}^{n} \mu_i B_{i,n}(x), \quad \text{with } \mu_0, \mu_1, \ldots, \mu_n \in \mathbb{R}.$$

- Hence, every polynomial (in power basis) can be seen as a Bernstein polynomial, albeit with unknown scalars for the linear combination.
- Can we select μ_i such that a decent approximation of a user-specified function is achieved?

Definition 197 (Bernstein approximation)

Consider a continuous function $f\colon [0,1] \to \mathbb{R}$. For $n \in \mathbb{N}$, the *Bernstein approximation* with degree n of f is defined as

$$B_{n,f}(x) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_{i,n}(x).$$

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 197 (Bernstein approximation)

Consider a continuous function $f\colon [0,1] \to \mathbb{R}$. For $n \in \mathbb{N}$, the *Bernstein approximation* with degree n of f is defined as

$$B_{n,f}(x) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_{i,n}(x).$$

- Hence, a Bernstein approximation is given by a Bernstein polynomial, with weights $\mu_i := f\left(\frac{i}{n}\right)$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Definition 197 (Bernstein approximation)

Consider a continuous function $f\colon [0,1] \to \mathbb{R}$. For $n \in \mathbb{N}$, the *Bernstein approximation* with degree n of f is defined as

$$B_{n,f}(x) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_{i,n}(x).$$

- Hence, a Bernstein approximation is given by a Bernstein polynomial, with weights $\mu_i := f\left(\frac{i}{n}\right)$.

Theorem 198 (Weierstrass 1885, Bernstein 1911)

The Bernstein approximation $B_{n,f}$ converges uniformly to the continuous function f on the interval $[0,1]$. That is, given a tolerance $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that

$$|f(x) - B_{n,f}(x)| \leq \varepsilon \qquad \text{for all } x \in [0,1] \text{ and all } n \geq n_0.$$

Bernstein Approximation

Definition 197 (Bernstein approximation)

Consider a continuous function $f\colon [0,1] \to \mathbb{R}$. For $n \in \mathbb{N}$, the *Bernstein approximation* with degree n of f is defined as

$$B_{n,f}(x) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) B_{i,n}(x).$$

- Hence, a Bernstein approximation is given by a Bernstein polynomial, with weights $\mu_i := f\left(\frac{i}{n}\right)$.

Theorem 198 (Weierstrass 1885, Bernstein 1911)

The Bernstein approximation $B_{n,f}$ converges uniformly to the continuous function f on the interval $[0,1]$. That is, given a tolerance $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that

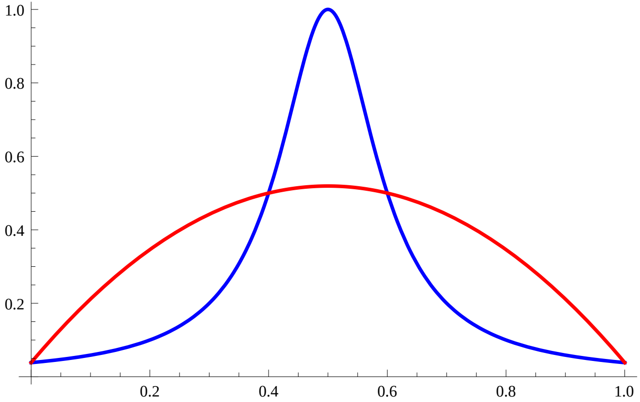
$$|f(x) - B_{n,f}(x)| \leq \varepsilon \qquad \text{for all } x \in [0,1] \text{ and all } n \geq n_0.$$

- Since $x := \frac{t-a}{b-a}$ maps $t \in [a,b]$ to $x \in [0,1]$, this approximation theorem extends to continuous functions $f\colon [a,b] \to \mathbb{R}$.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Sample Bernstein approximation of a continuous function:

$f\colon [0,1] \to \mathbb{R}$ $\qquad$ $f(x) := \dfrac{1}{1+(10x-5)^2}$



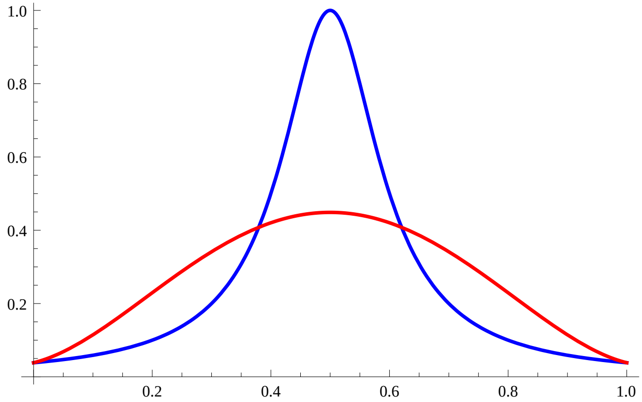
Degree 2

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

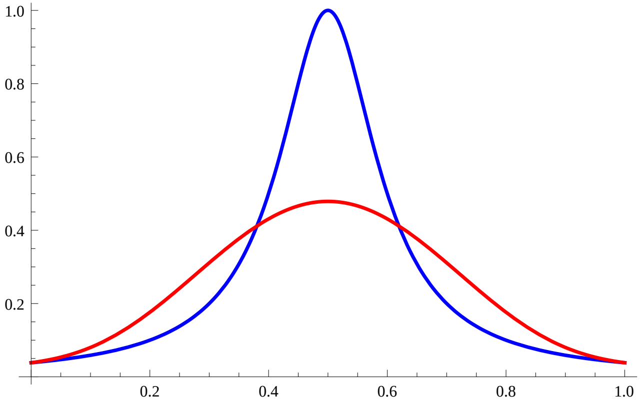
$$f\colon [0,1] \to \mathbb{R} \qquad\qquad f(x) := \frac{1}{1+(10x-5)^2}$$



Degree 4

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \frac{1}{1+(10x-5)^2}$$

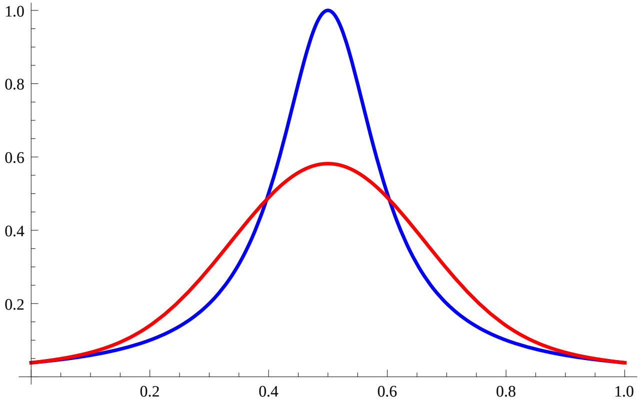


Degree 8

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

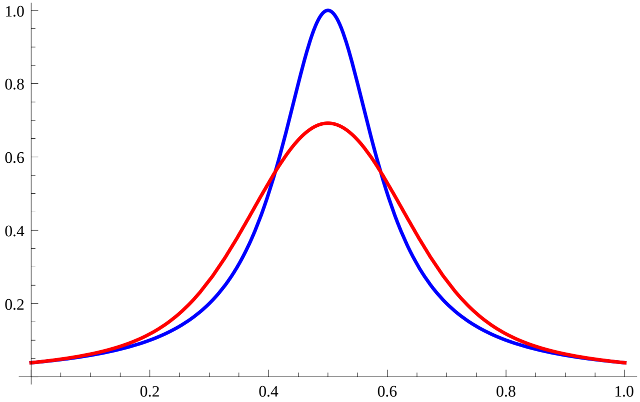
$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \frac{1}{1+(10x-5)^2}$$



Degree 16

- Sample Bernstein approximation of a continuous function:

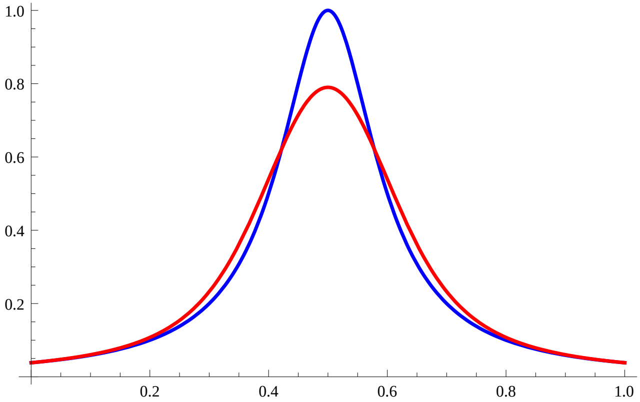
$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \frac{1}{1+(10x-5)^2}$$



Degree 32

- Sample Bernstein approximation of a continuous function:

$f\colon [0,1] \to \mathbb{R}$ $\qquad$ $f(x) := \dfrac{1}{1+(10x-5)^2}$

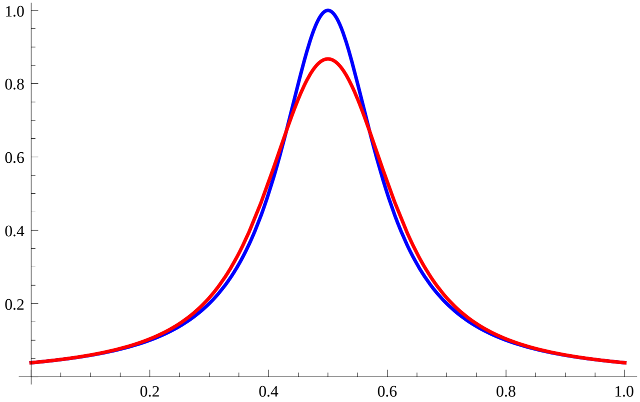


Degree 64

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

$f\colon [0,1] \to \mathbb{R}$ $\qquad$ $f(x) := \dfrac{1}{1+(10x-5)^2}$

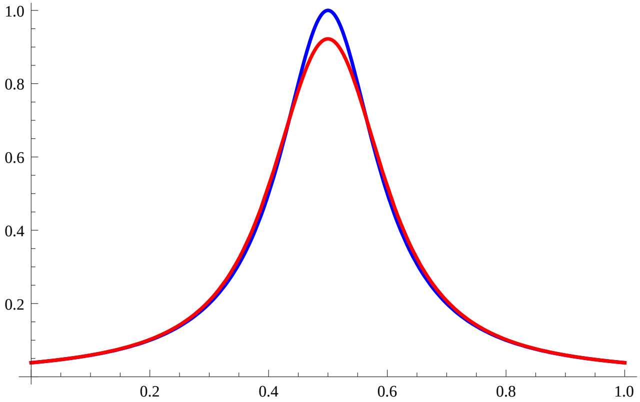


Degree 128

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \frac{1}{1+(10x-5)^2}$$

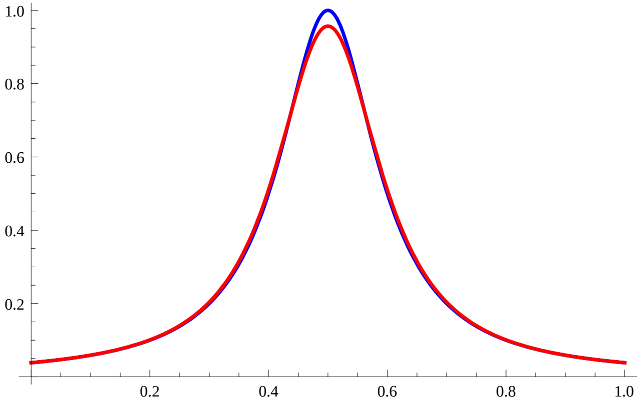


Degree 256

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \frac{1}{1+(10x-5)^2}$$

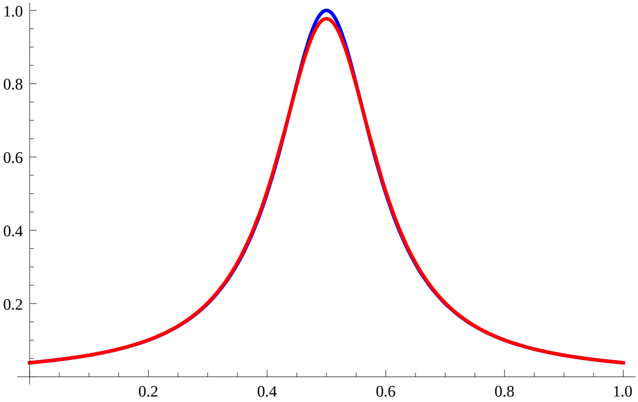


Degree 512

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \frac{1}{1+(10x-5)^2}$$

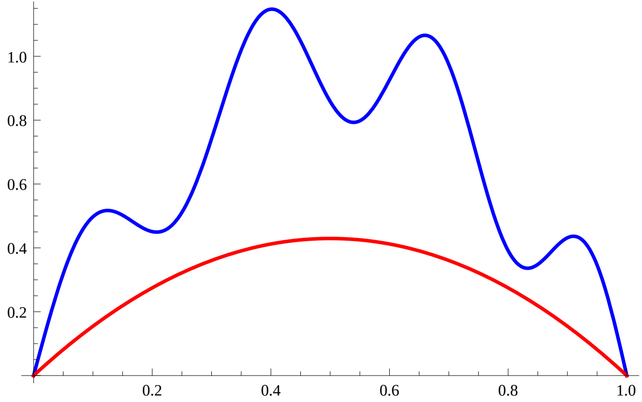


Degree 1024

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$

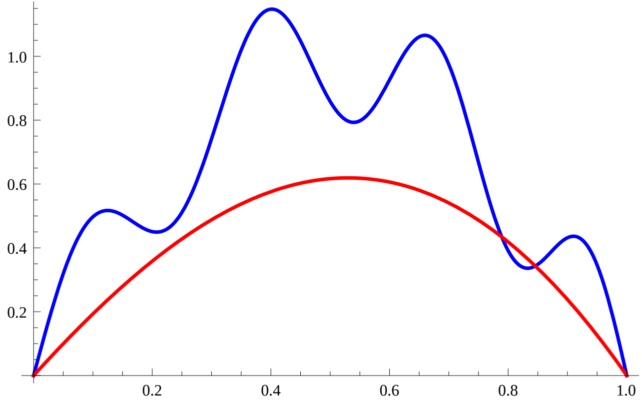


Degree 2

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

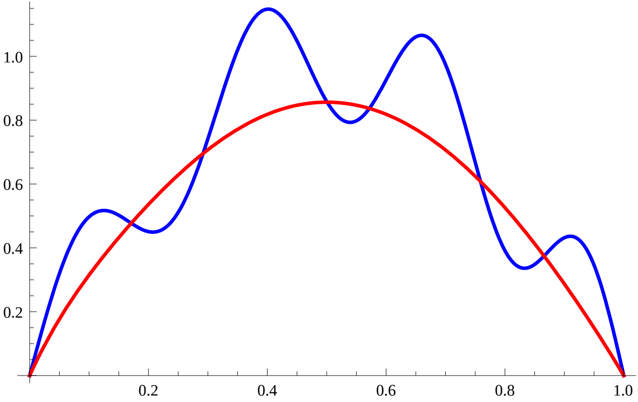
$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$



Degree 4

- Sample Bernstein approximation of a continuous function:

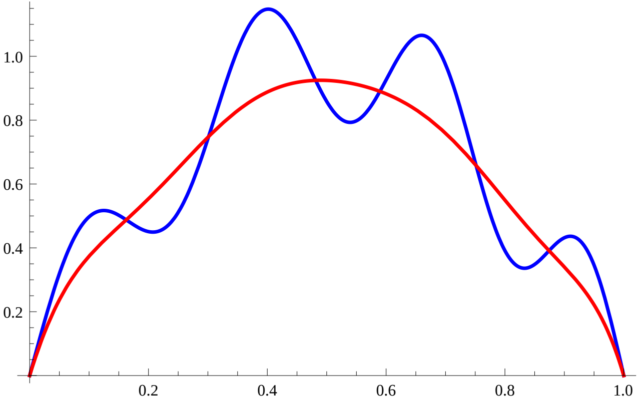
$$f\colon [0,1] \to \mathbb{R} \qquad\qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$



Degree 8

- Sample Bernstein approximation of a continuous function:

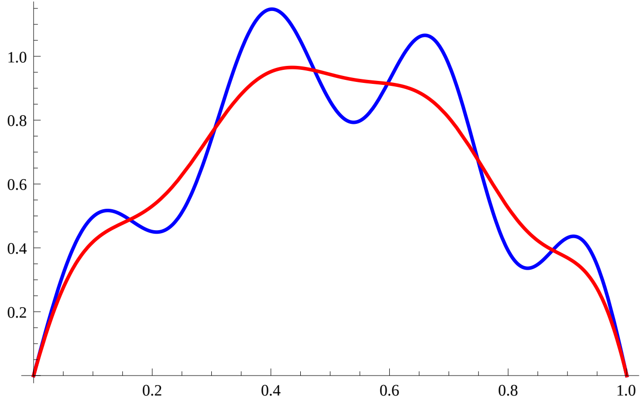
$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$



Degree 16

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad\qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$

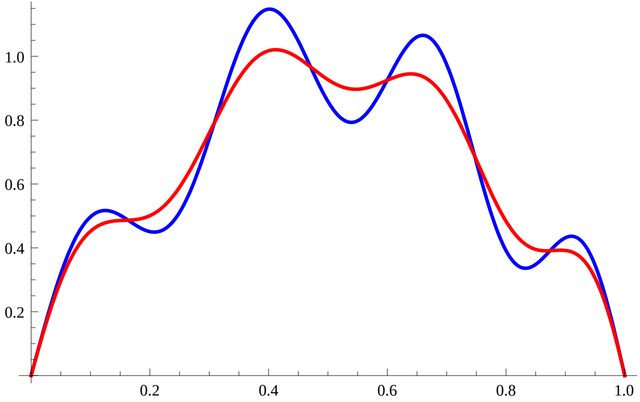


Degree 32

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad\qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$

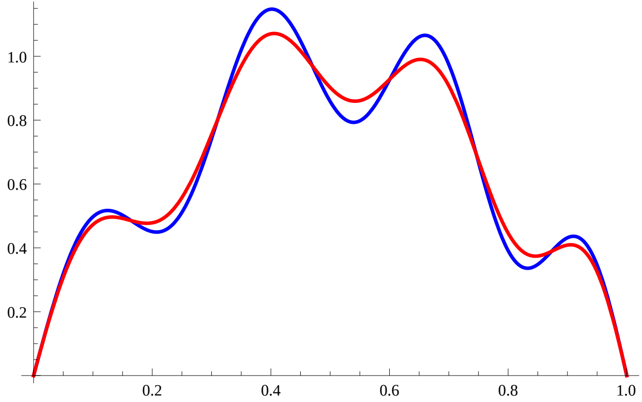


Degree 64

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

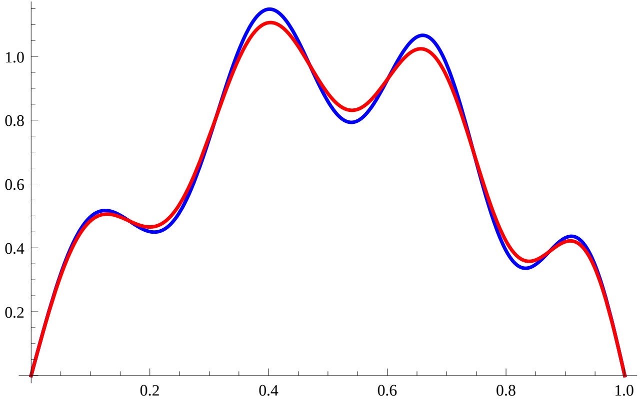
$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$



Degree 128

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$

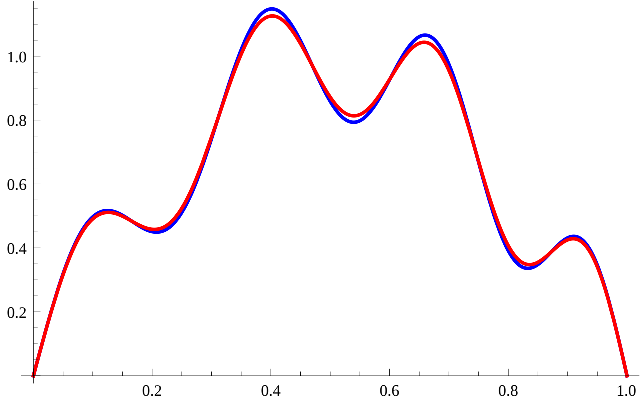


Degree 256

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$

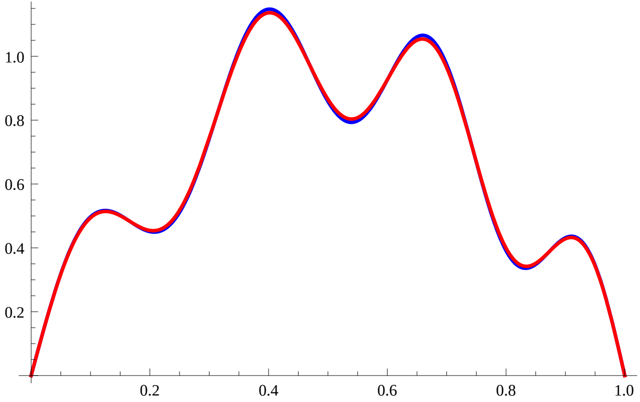


Degree 512

Sample Bernstein Approximation

- Sample Bernstein approximation of a continuous function:

$$f\colon [0,1] \to \mathbb{R} \qquad f(x) := \sin(\pi x) + \frac{1}{5}\sin\left(6\pi x + \pi x^2\right)$$



Degree 1024

The End!

I hope that you enjoyed this course, and I wish you all the best for your future studies.

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab