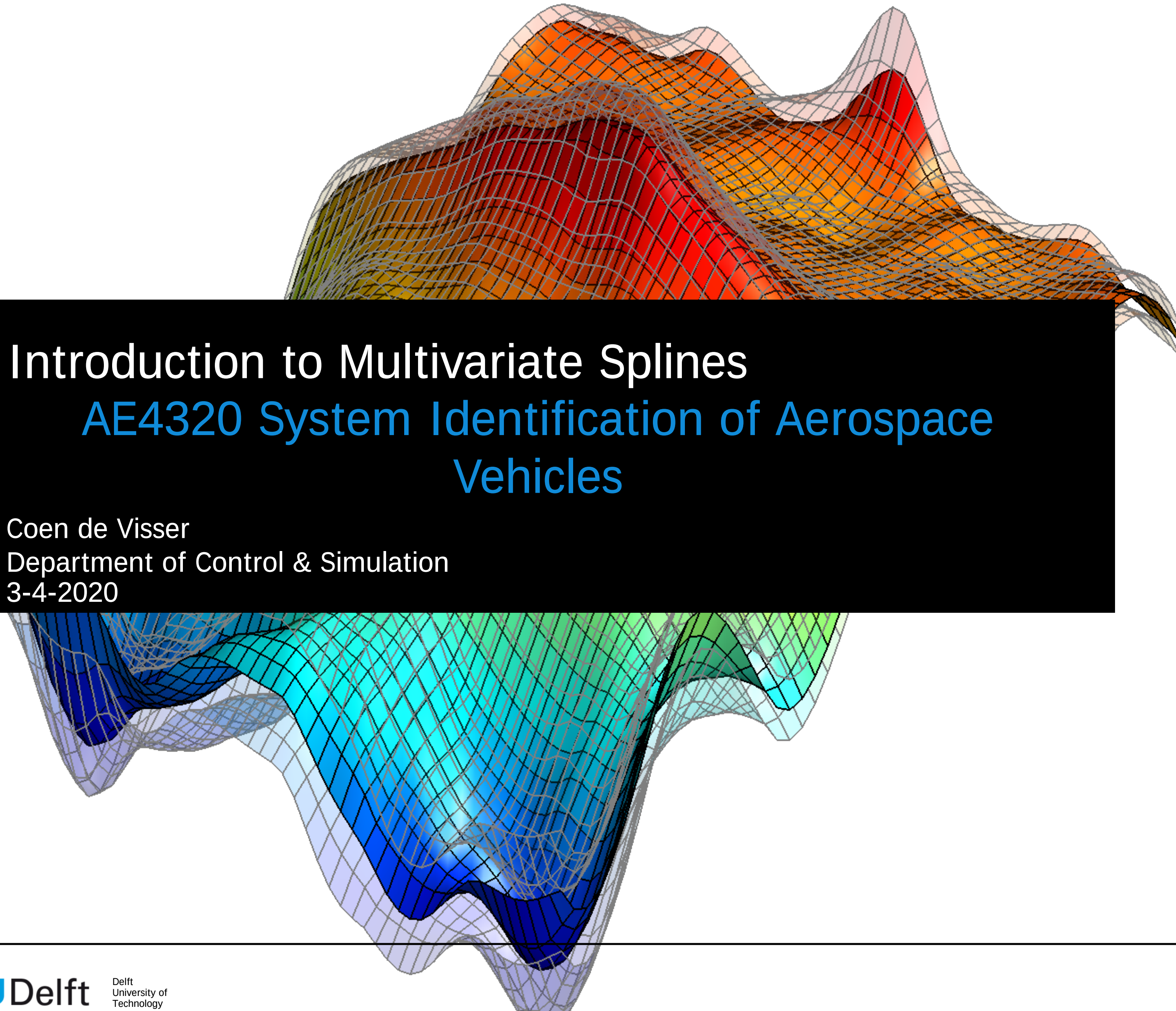


Introduction to Multivariate Splines

AE4320 System Identification of Aerospace Vehicles

Coen de Visser
Department of Control & Simulation
3-4-2020

Course Outline

- **Lecture 1: (dr.ir. Coen de Visser)**
  - Course goals and objectives
  - Introduction to System Identification

- **Lecture 2,3: (dr.ir. Daan Pool)**
  - System Identification Experiments

- **Lecture 4,5,6: (dr.ir. Daan Pool)**
  - Kalman filters
  - State estimation & Sensor Fusion

- **Lecture 7,8: (dr.ir. Coen de Visser)**
  - Model structure selection
  - Model parameter estimation

Course Outline

- **Lecture 9: (dr.ir. Coen de Visser)**
  - Advanced identification approach: Neural networks
- **Lecture 10,11: (dr.ir. Coen de Visser)**
  - Advanced identification approach: Multivariate B-Splines
- **Lecture 12: (dr.ir. Coen de Visser)**
  - Model validation, course conclusion

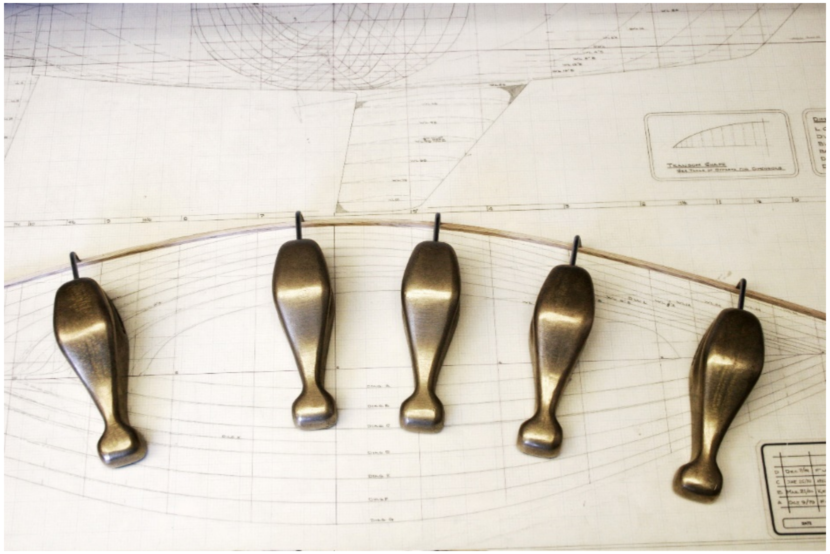
This Lecture

1. Introduction
2. Polynomials and Splines
3. Simplex Polynomials
4. Ordinary least squares estimation with Simplex Polynomials
5. Introducing multivariate Simplex B-splines
6. Continuity of multivariate Simplex B-splines
7. In-depth discussion on continuity of multivariate Simplex B-Splines
8. Estimation of Simplex B-splines
9. Advanced applications of simplex B-splines
10. Current spline research & Conclusion

Goals of this Lecture

Questions that will be answered during this lecture:

1. *What is the relation between a spline and a polynomial?*
2. *What are the different types of multivariate splines?*
3. *What is a multivariate simplex B-spline?*
4. *What are the unique properties of multivariate simplex B-splines?*
5. *How can we use simplex B-splines in aerospace applications?*



Introduction

Where we are now in the System Identification Cycle:

Experiment phase

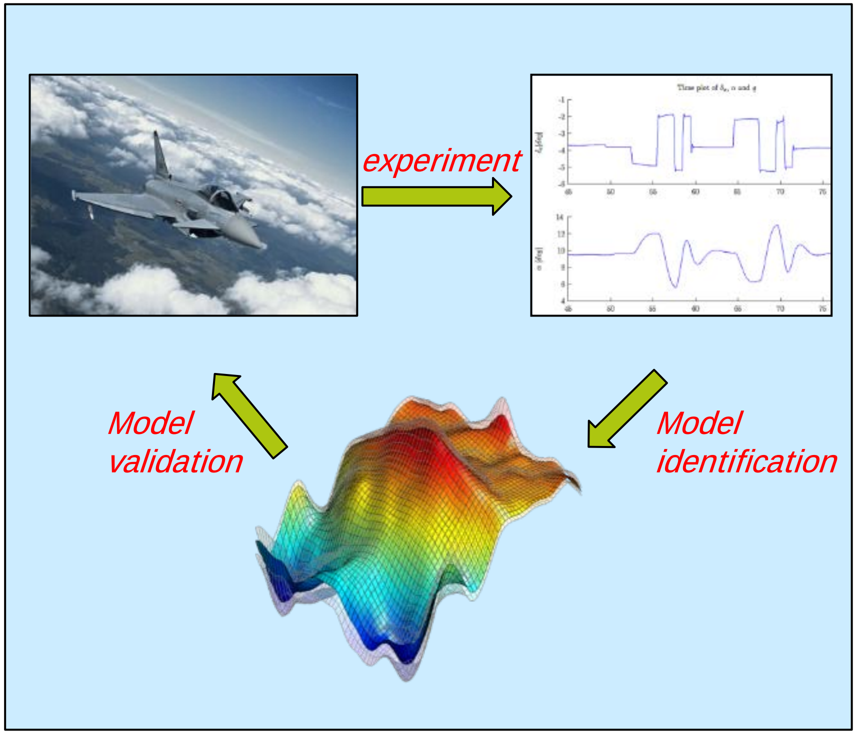
- Plant analysis
- Experiment design and execution
- Data logging and pre-processing

Model identification phase

- State estimation
- Model structure definition
- Parameter estimation

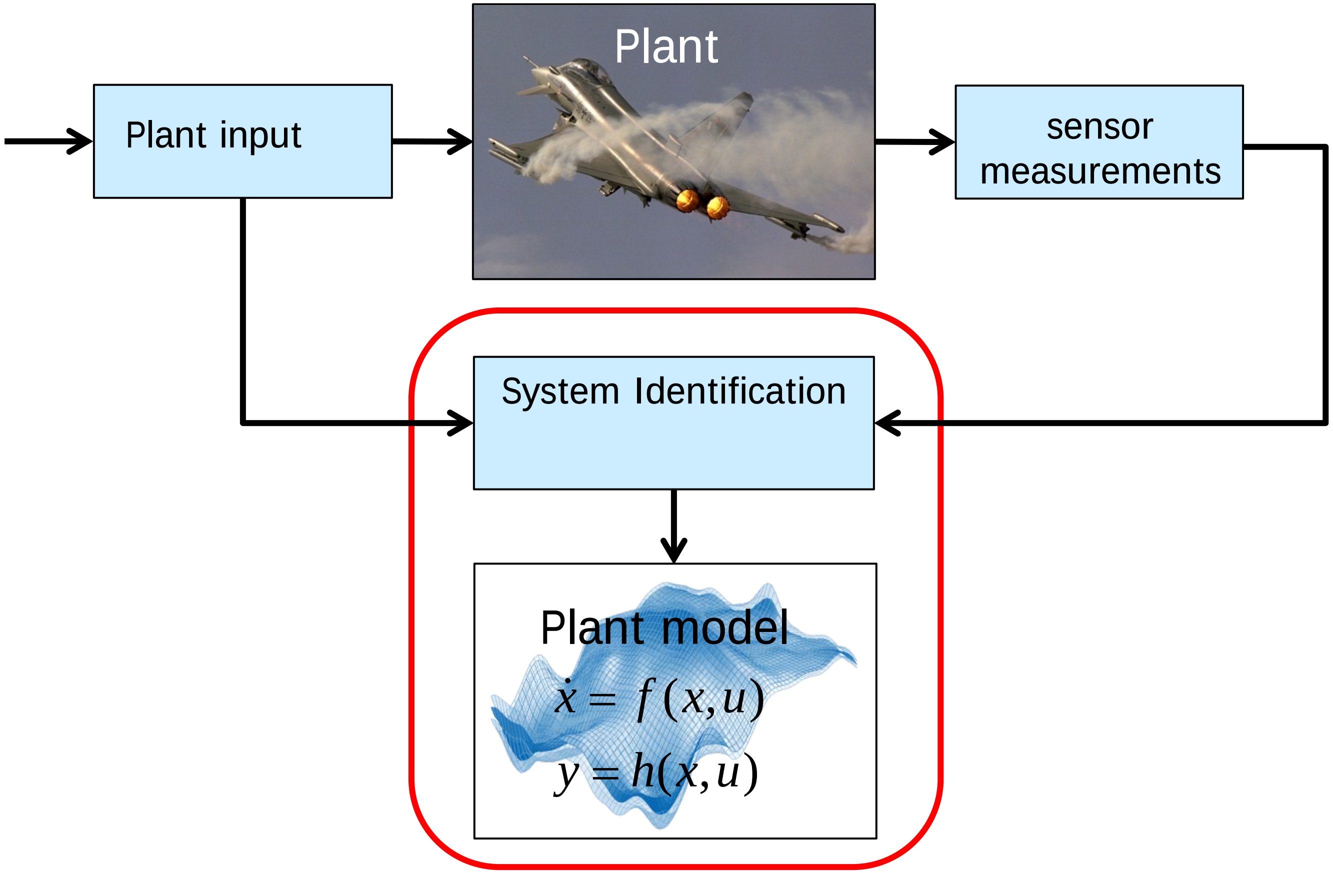
Model validation phase

- Model validation



Introduction

The focus is now on the modelling part of the SysID procedure...



Introduction

Aerodynamic models: real life

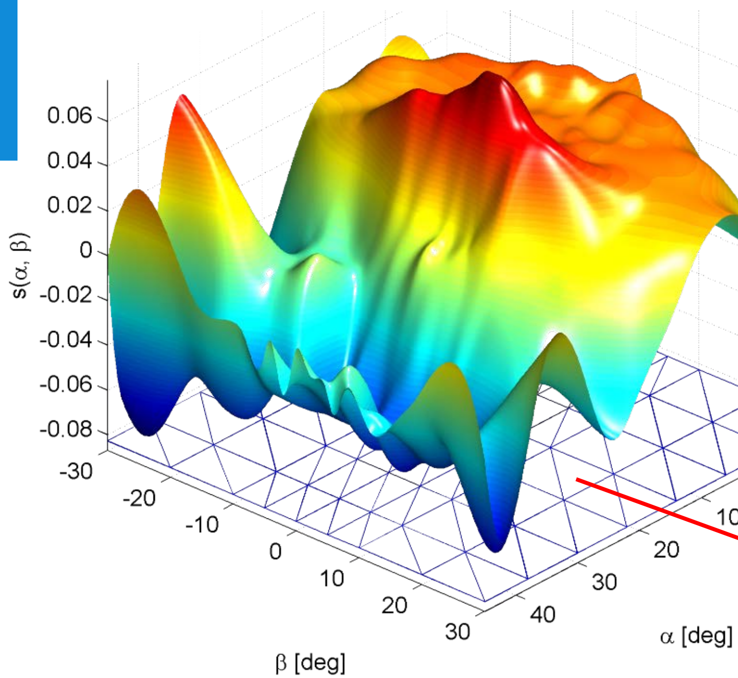
- In real life aerodynamic models are extremely complex and expensive to create.
- Example: Eurofighter Typhoon ADM
  - 260 multidimensional data tables
  - >20 years in the making
  - More than 5000 test flights
  - Single test flight costs around 100.000 euros!

Introduction

Aerodynamic Model Identification at Eurofighter

- **Linear model structures, Bayesian parameter estimation.**
- **Problem: highly nonlinear aerodynamics incompatible with linear models**
- **Solution at the time: manual smoothing…!**
- **A General solution: Multivariate Splines**

Introduction

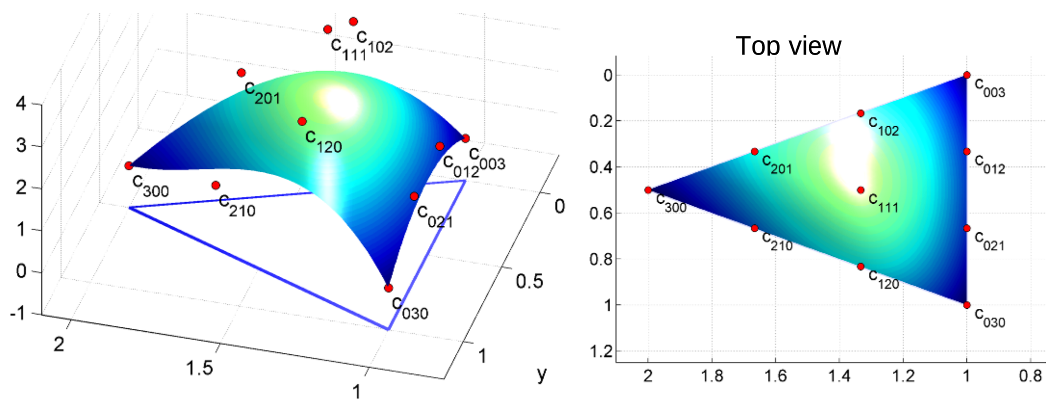


Spline model: F-16 leading edge flap deflection effect on pitching moment.

Properties Multivariate Simplex Splines:

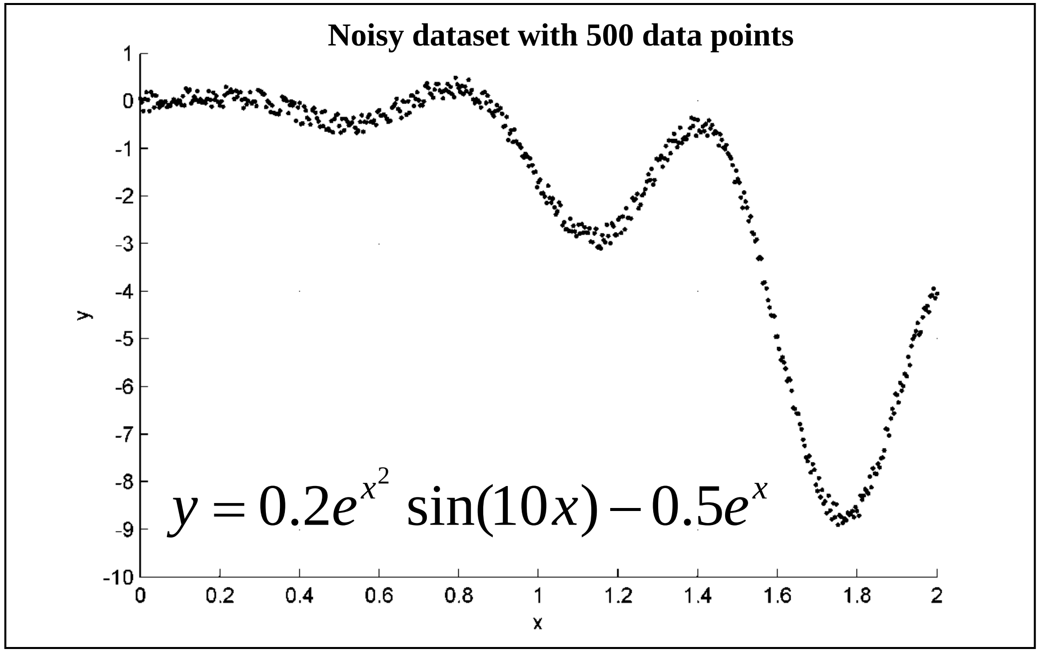
- **high approximation power**.
- spline continuity order can be chosen freely.
- spline parameters have a spatial location allowing **local model modification**.
- naturally capable of fitting **scattered** datasets on **non-rectangular domains**.

Single simplex (triangle) with bivariate polynomial



Polynomials & Splines

Fitting data with high order polynomials



Weierstrass Approximation Theorem (1885)

Every continuous function defined on an interval [a, b] can be approximated *as closely as desired* by a polynomial function.

Polynomials & Splines

Least squares data fitting with high order polynomials

Assume the following univariate polynomial model structure:

$$
\begin{aligned}
y(i) &= p\left(x,\theta\right) + \varepsilon(i) \\
&= \theta_0 + \theta_1 x(i) + \theta_2 x^2(i) + \cdots + \theta_n x^n(i) + \varepsilon(i)
\end{aligned}
$$

In Matrix form:

$$
y(i) = \begin{bmatrix} 1 & x(i) & x^2(i) & \cdots & x^n(i) \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} + \varepsilon(i)
$$

Polynomials & Splines

Least squares data fitting with high order polynomials

Matrix form for all N measurements:

$$Y = \begin{bmatrix} 1 & x(1) & x^2(1) & \cdots & x^M(1) \\ 1 & x(2) & x^2(2) & \cdots & x^M(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x(N) & x^2(N) & \cdots & x^M(N) \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

$$= A(x) \cdot \theta + \varepsilon$$

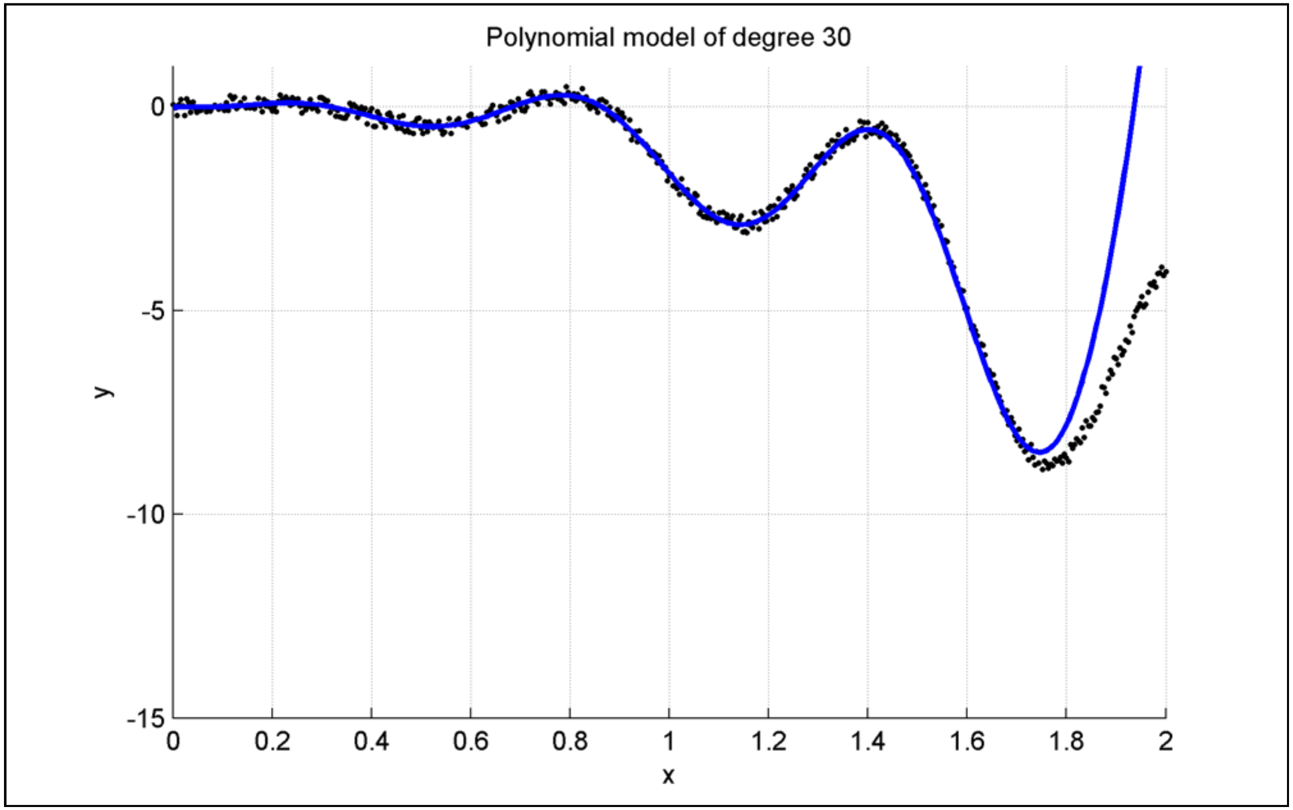
Recall the least squares estimator for θ:

$$\hat{\theta}_{OLS} = \left(A^T(x) \cdot A(x) \right)^{-1} \cdot A^T(x) \cdot Y$$

Polynomials & Splines

Weierstrass Approximation Theory in practice

Data fitting with high order polynomials



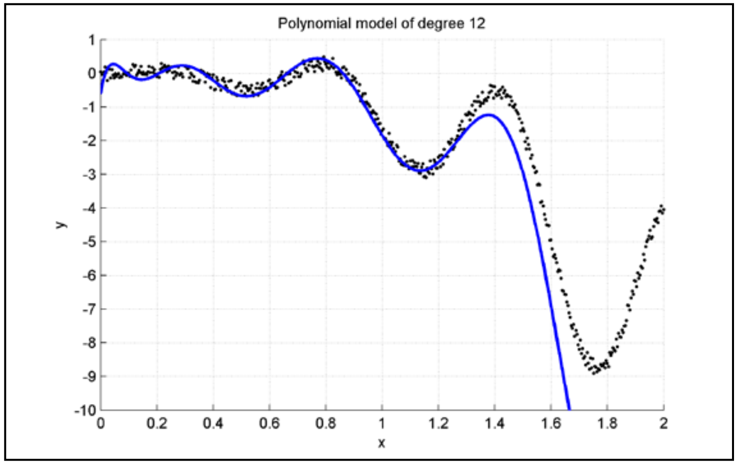
Polynomials & Splines

OLS data fitting with high order polynomials

Numerical Instability

The least squares regression $\boldsymbol{X}$ matrix tends to be badly scaled for high order polynomials. This is a direct result of the fact that ordinary polynomials do not have a stable basis...

$$y = 181x^{12} - 1724x^{11} + 6476x^{10} - 11125x^{9} + 4267x^{8} + 16672x^{7} - 32418x^{6} + 27563x^{5} + 12676x^{4} + 3154x^{3} - 391x^{2} + 20x + 0.25$$



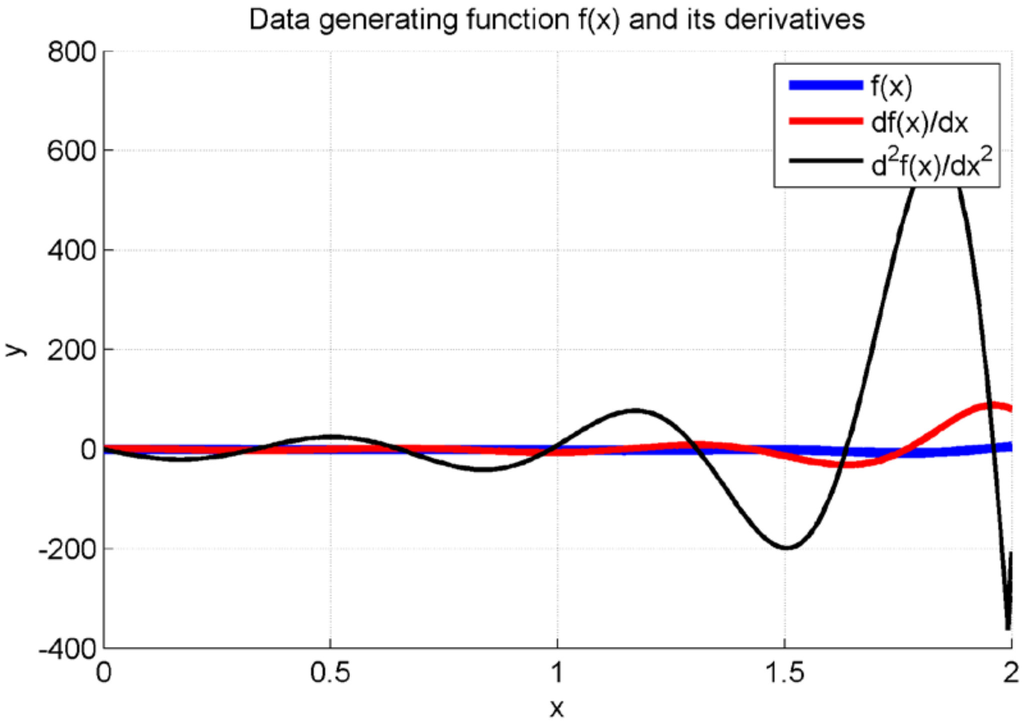
Polynomials & Splines

Runge's Phenomenon (1901)

A polynomial p of degree n approximating a continuous function f on [a, b] oscillates towards the end of the interval. These oscillations increase with the degree of the polynomial.

Reason:

The error between the generating function f and an interpolating polynomial of degree n is bounded by the n^{th} derivative of f, which increases in magnitude for most functions.



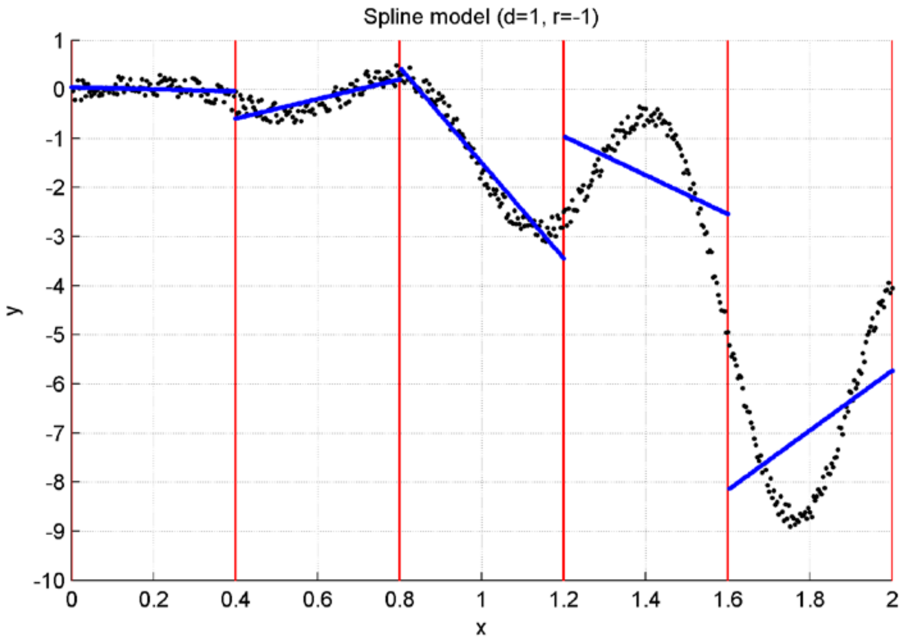
Polynomials & Splines

Solution:

1. Chop up the domain into smaller pieces
2. **for each piece**, fit polynomials of **lower degree**...

This circumvents Runge's phenomenon: by splitting the interval [a,b] into a number of subdomains, lower order polynomials can be used to accurately fit the data.

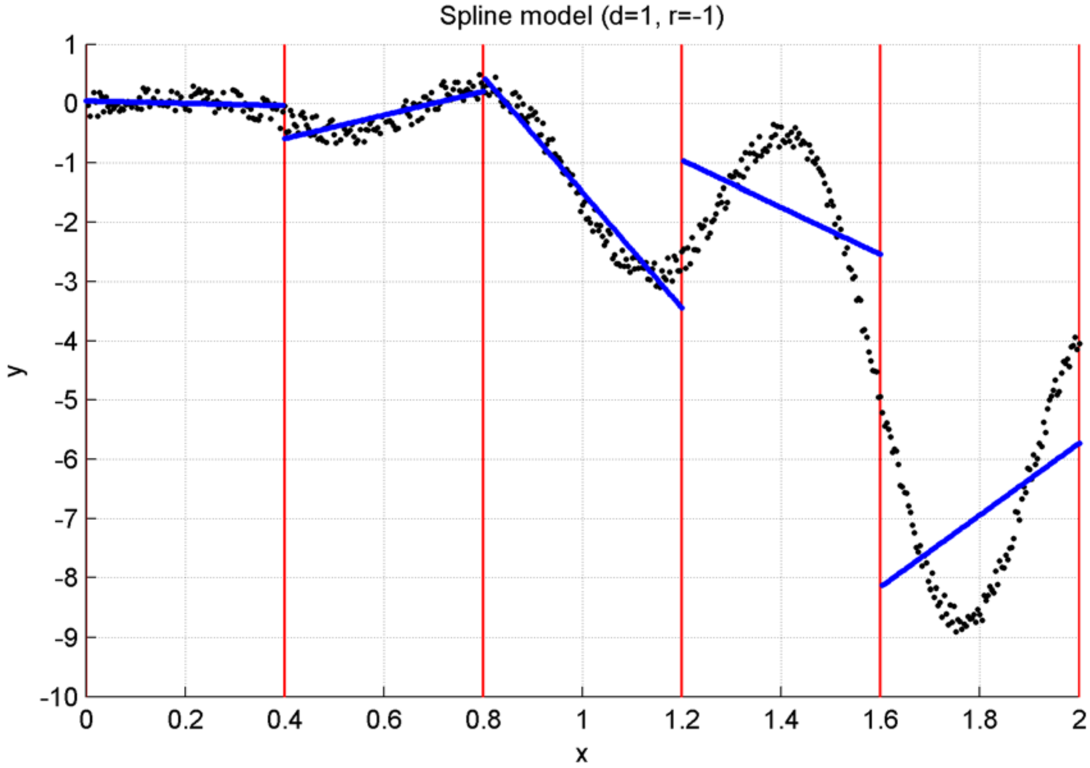
Lower order polynomials lead to **smaller errors on domain edges**...



[0, 2] -> {[0, 0.4], [0.4, 0.8], [0.8, 1.2], [1.2, 1.6], [1.6, 2.0]}

Polynomials & Splines

Polynomial splines: circumventing Runge's phenomenon…



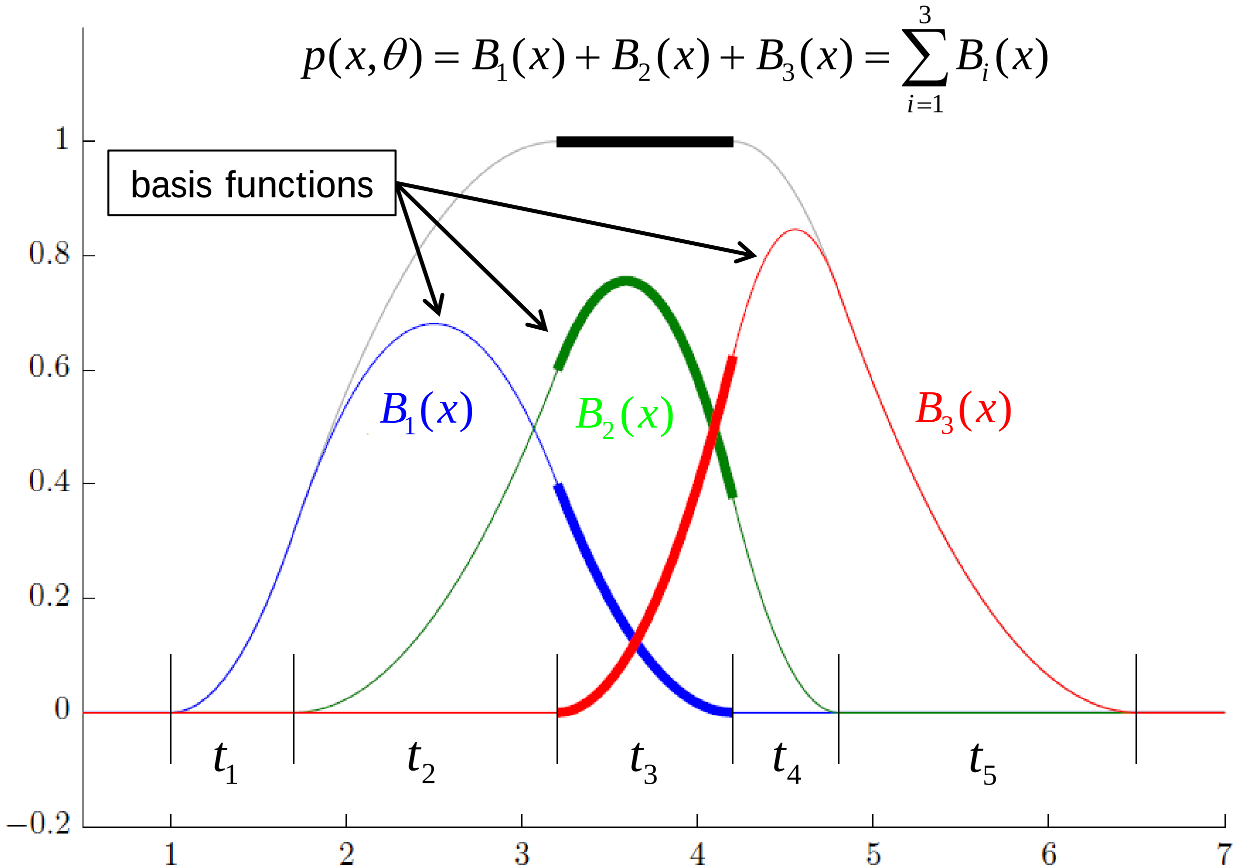
Spline

A Spline is a piecewise defined polynomial function with a predefined continuity between its pieces.

Polynomials & Splines

What is a “Basis Function”?

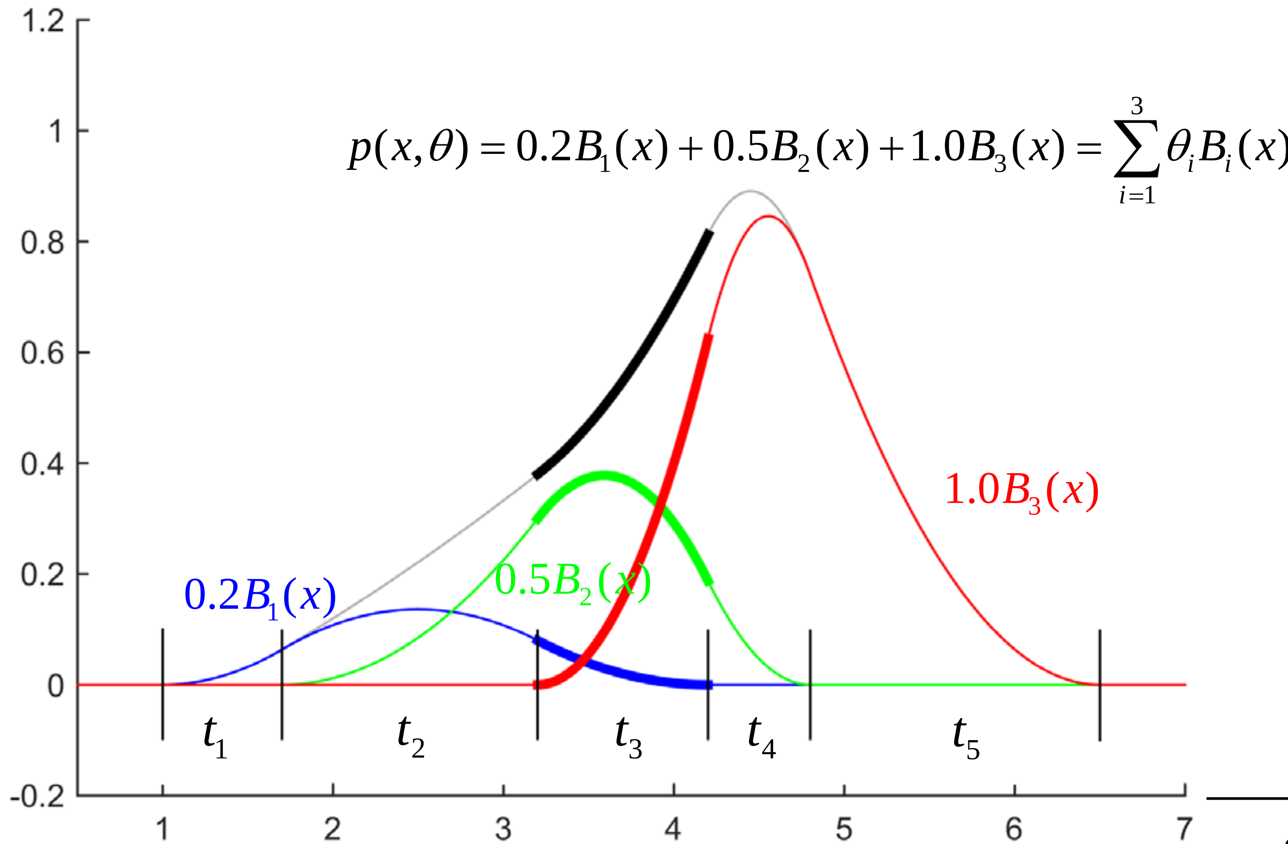
A basis function is a function (polynomial, Gaussian, Sine, Sigmoid, etc…) that when **summed** and **scaled**, can approximate any other function.



Polynomials & Splines

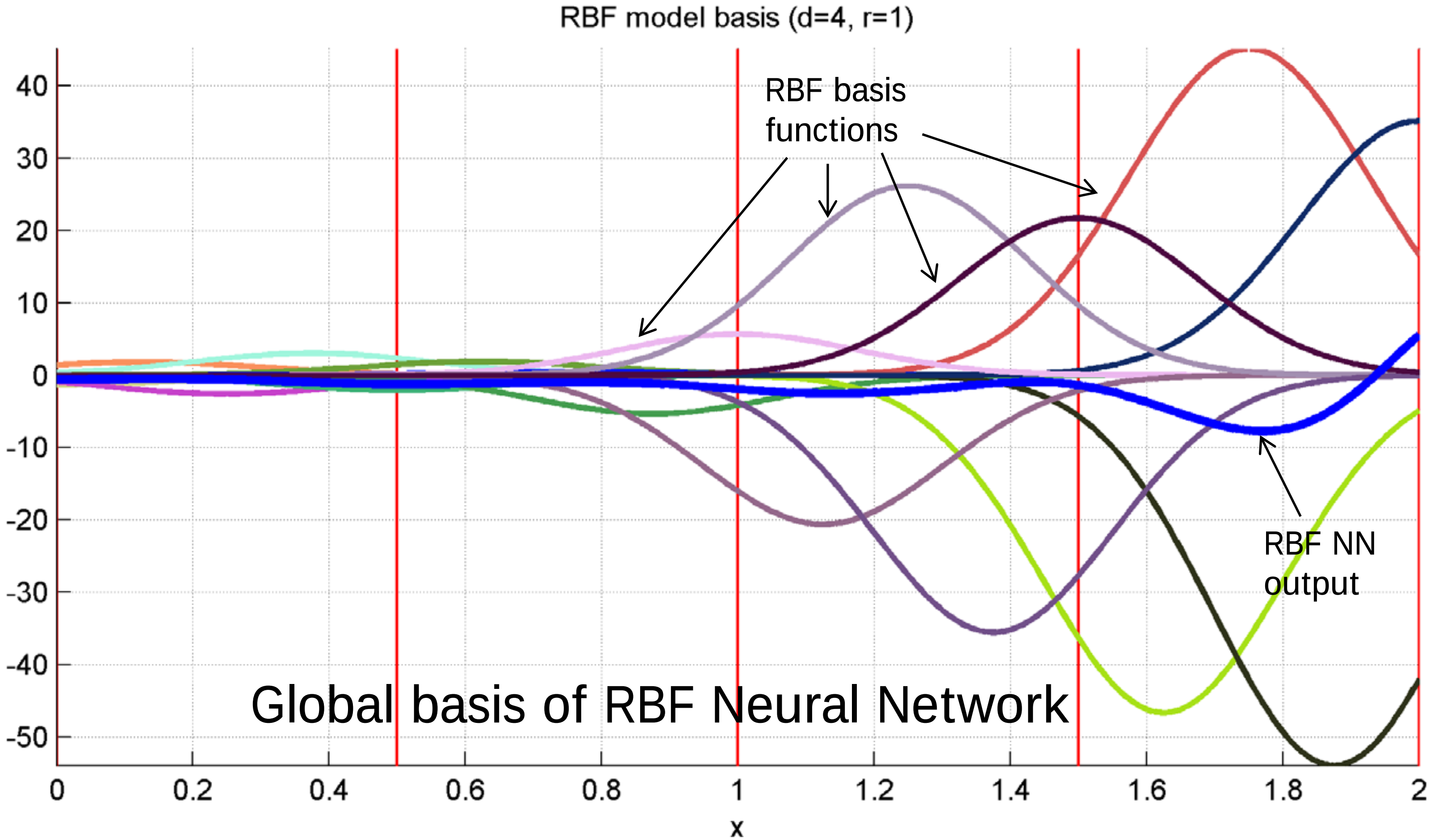
What is a "Basis Function"?

Each basis function scaled by a factor θ_i before summing: $p(x,\theta)=\sum_{i=1}^{M}\theta_i B_i(x)$



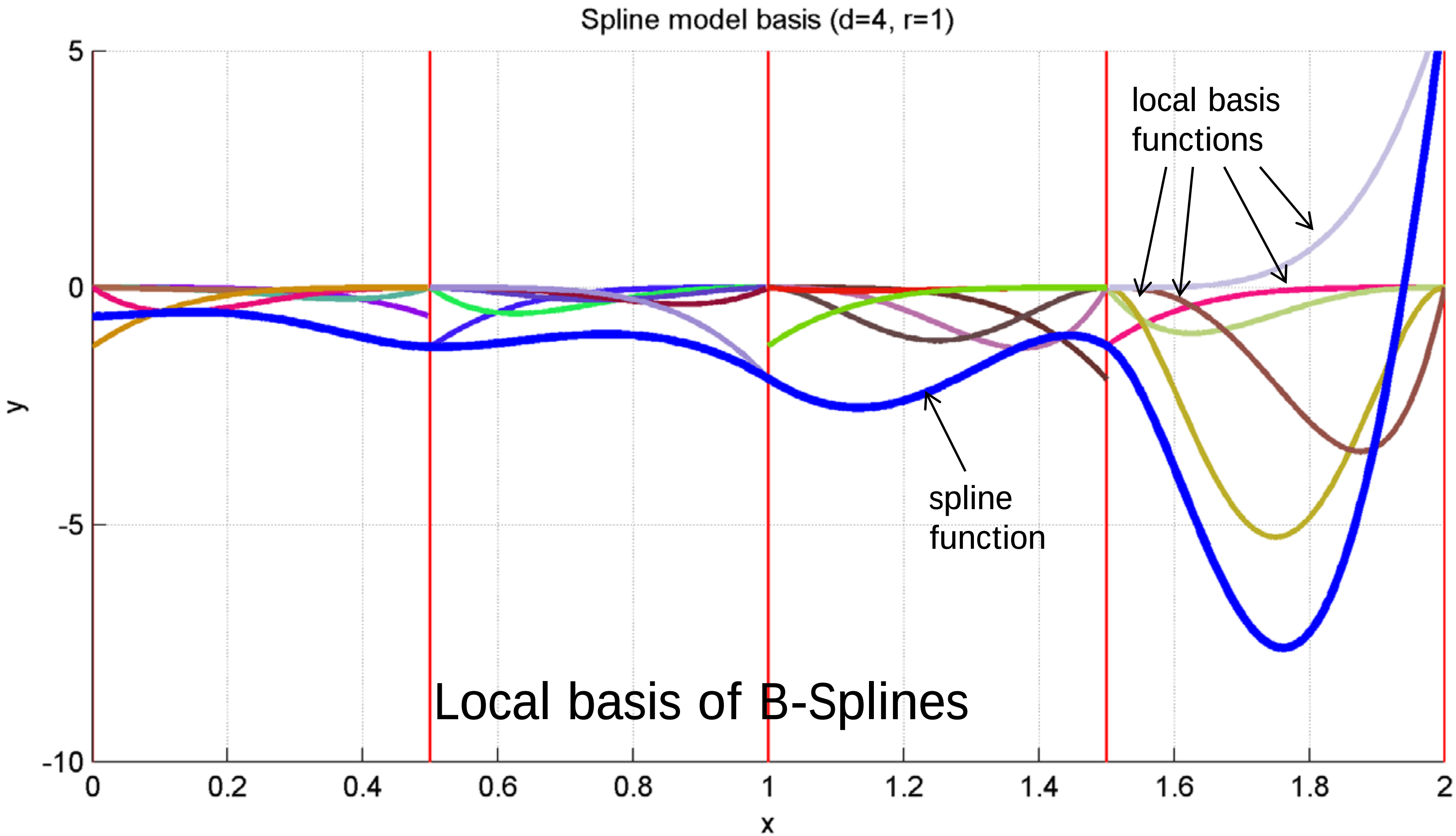
Polynomials & Splines

Local vs. Global basis functions



Polynomials & Splines

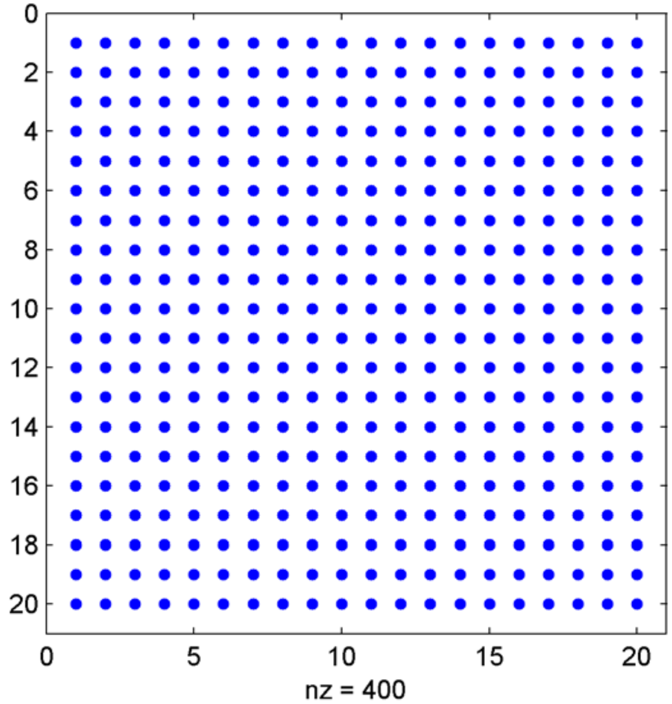
Local vs. Global basis functions



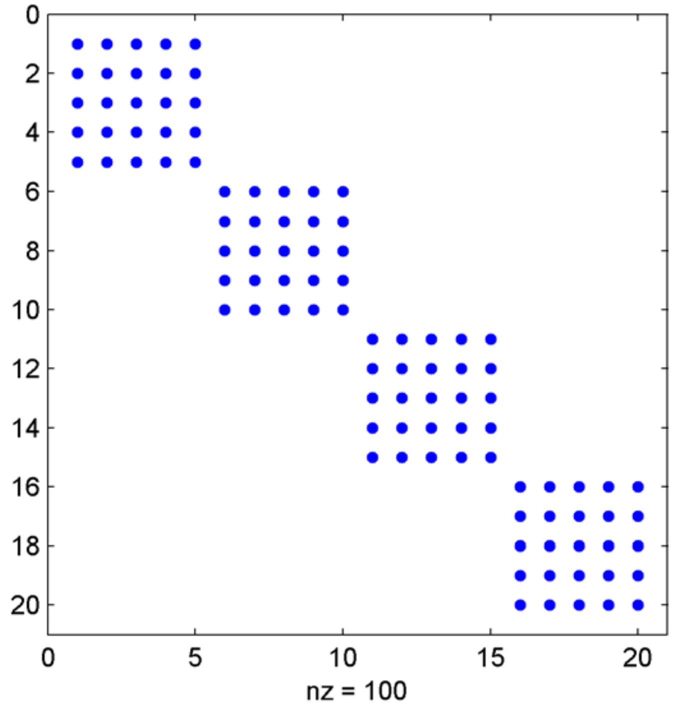
Local basis of B-Splines

Polynomials & Splines

Local vs. Global basis functions



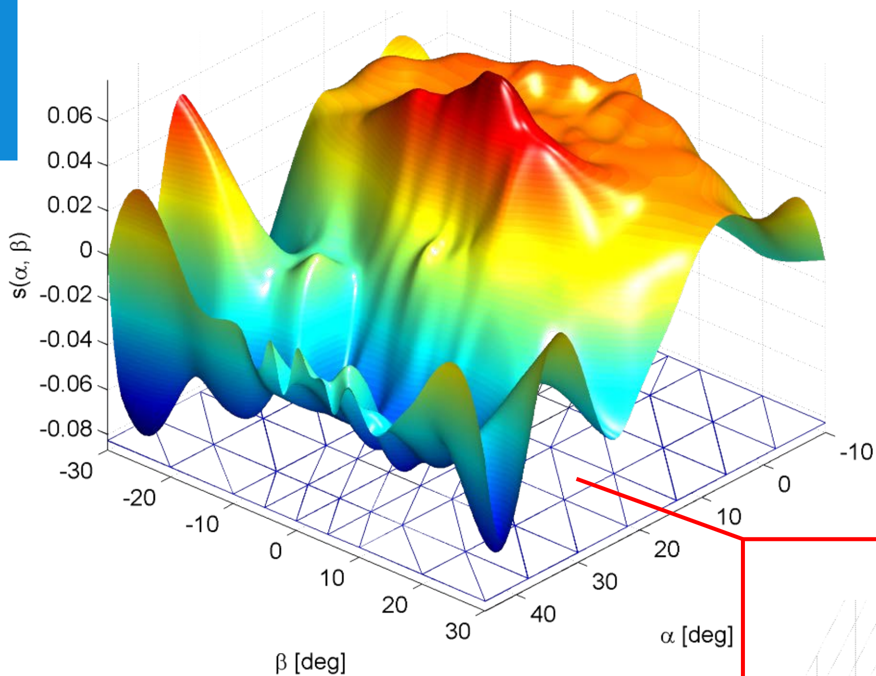
$A^T(x) \cdot A(x)$ for global basis



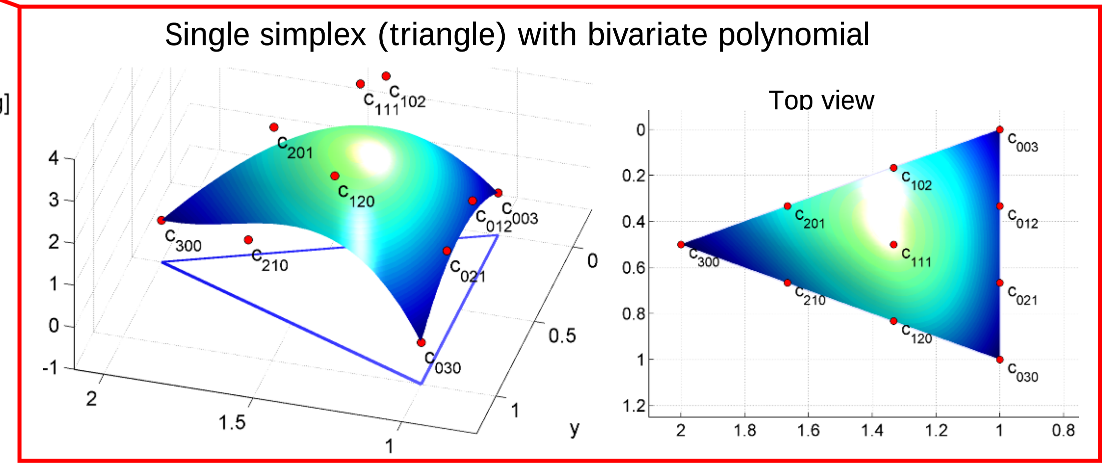
$A^T(x) \cdot A(x)$ for local basis

Sparsity directly leads to efficient (distributed) solvers and evaluators!

Simplex Polynomials



Spline model: F-16 leading edge flap deflection effect on pitching moment.



Simplex Polynomials

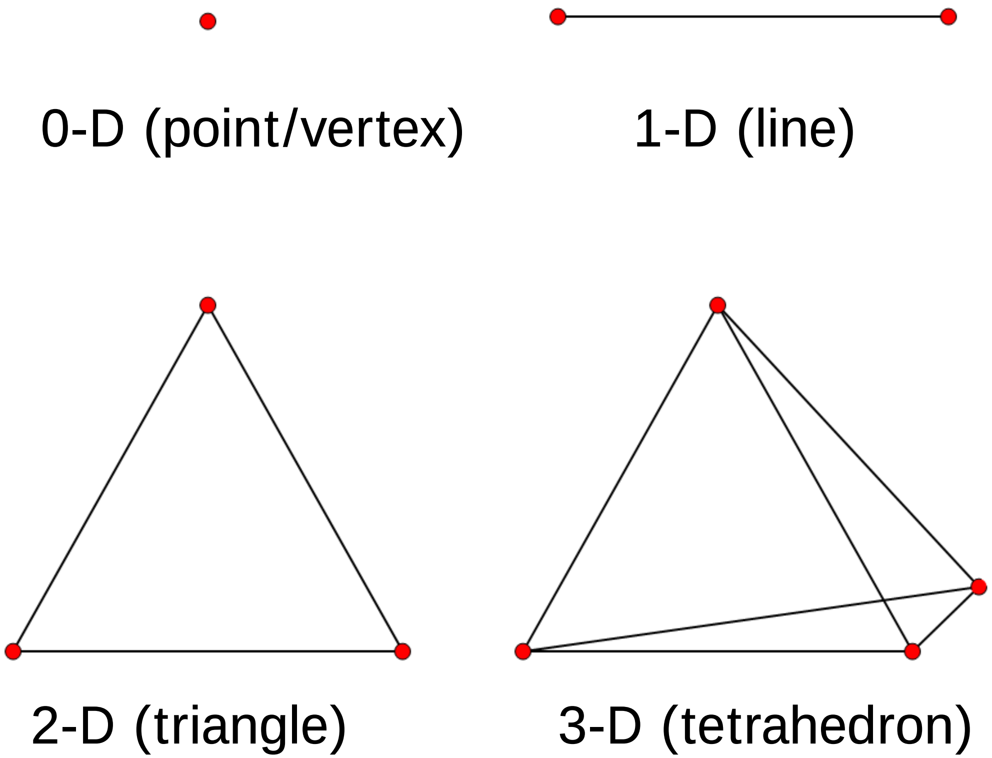
Multivariate Simplex Spline Polynomials are defined on Simplices:

The Simplex

A geometrical structure that minimally spans a set of dimensions.

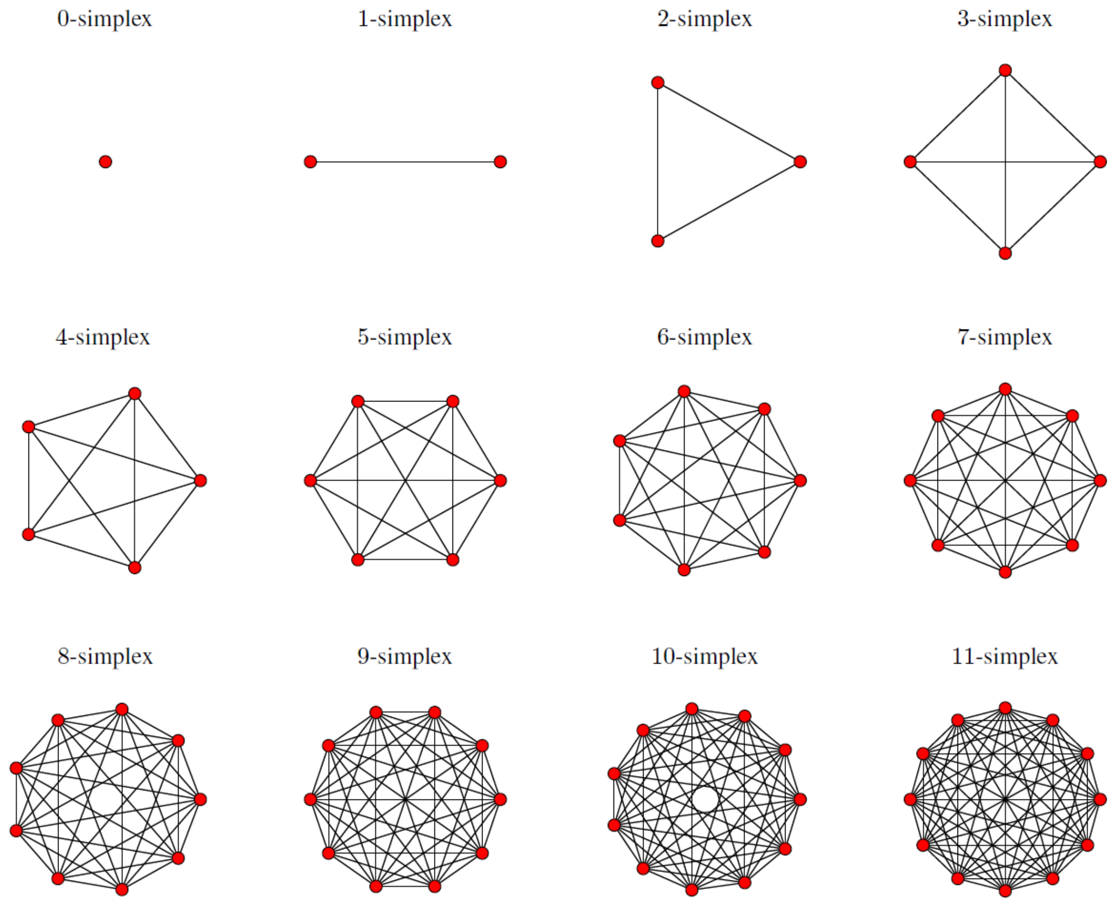
Rule:

The simplex of dimension *n* has *n*+1 non-degenerate vertices.



Simplex Polynomials

Multivariate Simplex Spline Polynomials are defined on Simplices:



The Simplex

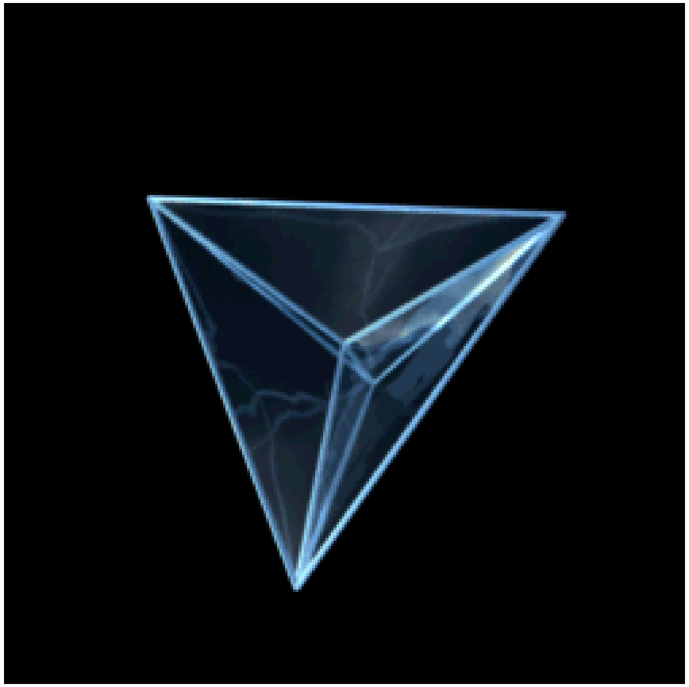
A geometrical structure that minimally spans a set of dimensions.

Rule:

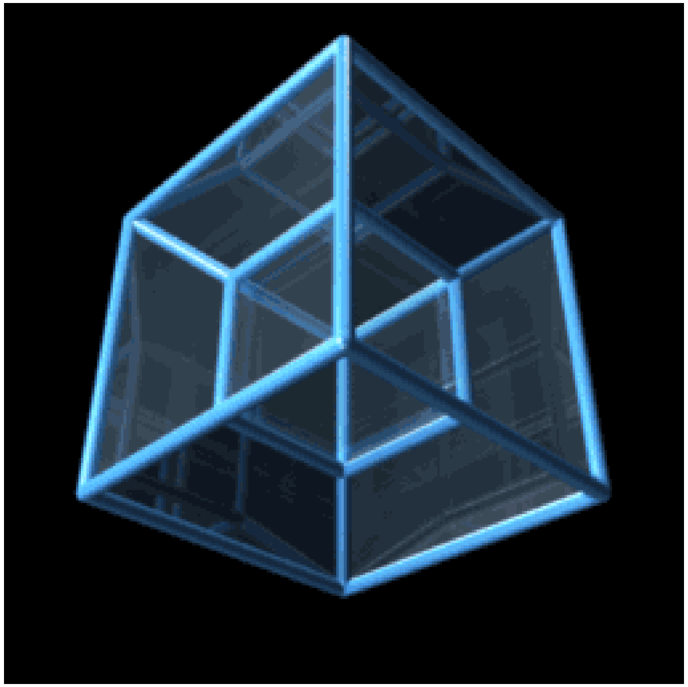
The simplex of dimension *n* has *n*+1 non-degenerate vertices.

Simplex Polynomials

High dimensional geometries can be difficult to visualize…



Pentachoron, the 4-simplex

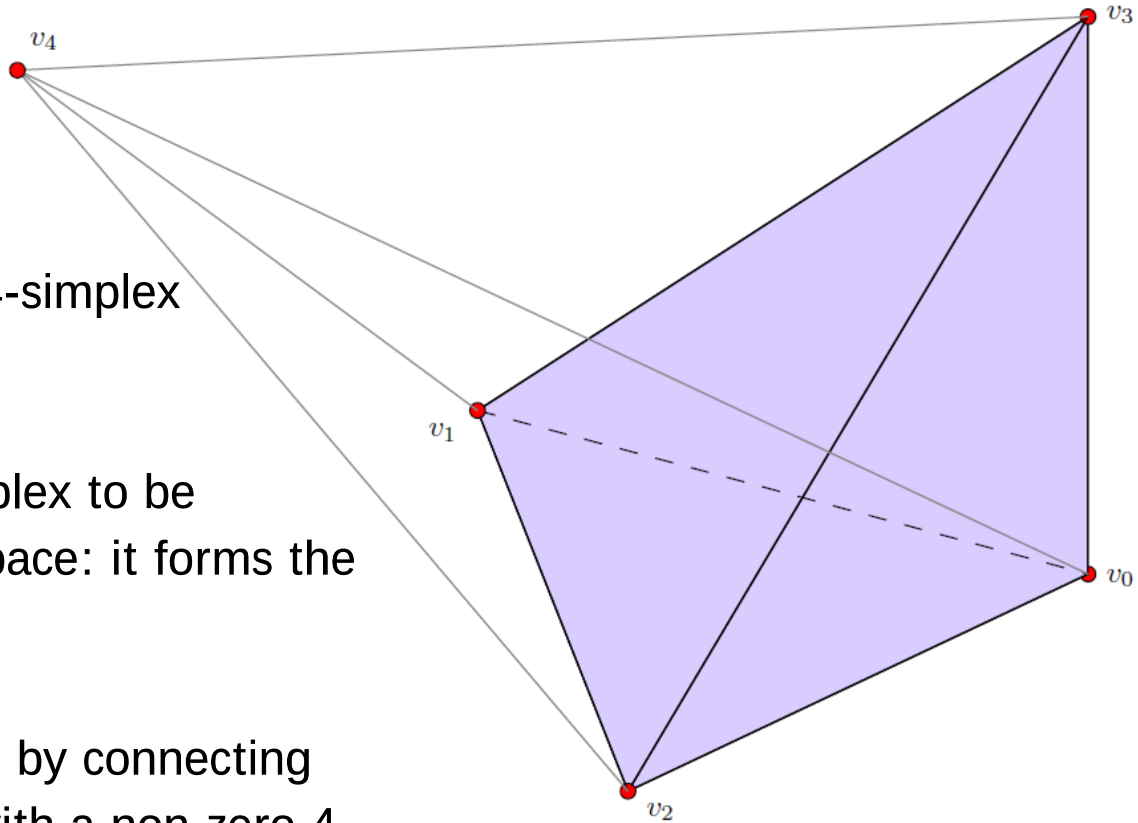


4-dimensional hypercube

Simplex Polynomials

High dimensional geometries can be difficult to visualize…

On the right is a so-called pentachoron, the 4-simplex.



Note that the 'edges' of the 4-simplex are tetrahedrons (3-simplex)!

Think of 1 edge of the 4-simplex to be completely embedded in 3-space: it forms the 'base' of the 4-simplex.

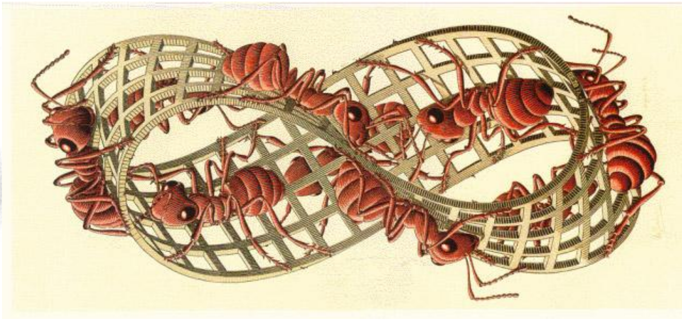
Then the 4-simplex is formed by connecting this base to a single vertex with a non-zero 4-space coordinate component…

Simplex Polynomials

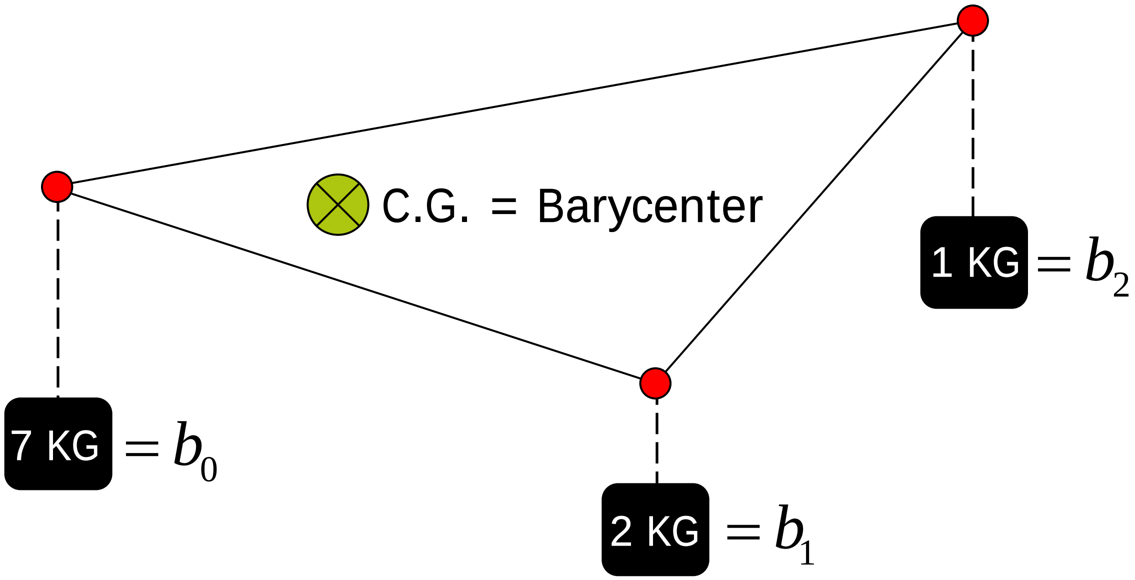
Simplices have a local coordinate system: **Barycentric coordinates**

Invented by German mathematician

August Ferdinand Möbius in 1827



Principle of Barycentric Coordinates:



Barycentric Coordinate of C.G. = (7, 2, 1)

Normalized Barycentric Coordinate of C.G. = (0.7, 0.2, 0.1)

Simplex Polynomials

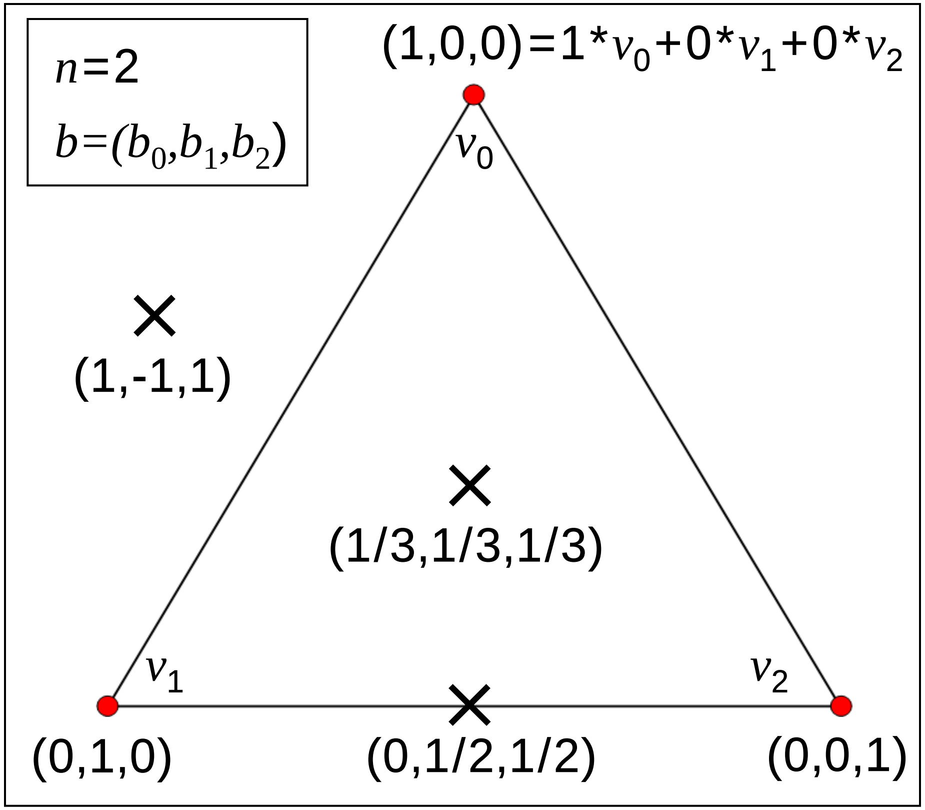
Simplices have a local coordinate system: **Barycentric coordinates**

The **unique** weights b are the barycentric coordinates of x w.r.t. the vertex coordinates v:

$$x = \sum_{i=0}^{n} b_i v_i$$

The barycentric coordinates are normalized:

$$1 = \sum_{i=0}^{n} b_i$$



Simplex Polynomials

The Cartesian to Barycentric coordinate transform

The Cartesian to Barycentric coordinate transform of the Cartesian coordinate x with respect to the n-simplex t_j is a linear transformation:

$$\begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^T = A_{t_j}^{-1} \cdot \left(x^T - v_0^T \right)$$

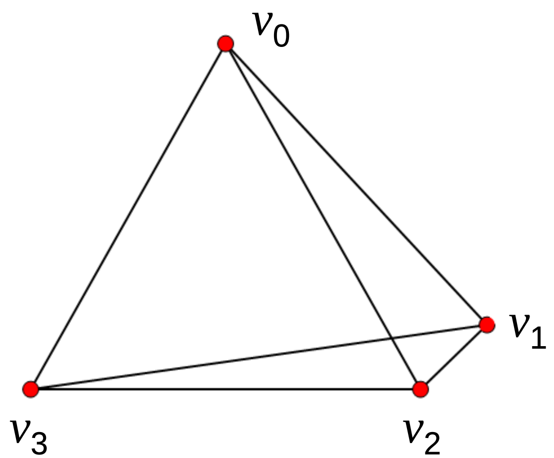
the normalization property is:

$$b_0 = 1 - \sum_{i=1}^{n} b_i$$

The non-singular matrix A_{t_j} given by:

$$A_{t_j} = \left[\left(v_1 - v_0\right)^T \quad \left(v_2 - v_0\right)^T \quad \cdots \quad \left(v_n - v_0\right)^T \right]$$

and $(v_0, v_1, \ldots, v_n)$ are the vertices of the n-simplex t_j.



In the following, we will use a short hand notation for the barycentric coordinate transformation:

$$\boxed{b = b_{t_j}(x),\ x \in t_j}$$

Simplex Polynomials

Example 6.4: Cartesian to Barycentric coordinate transform

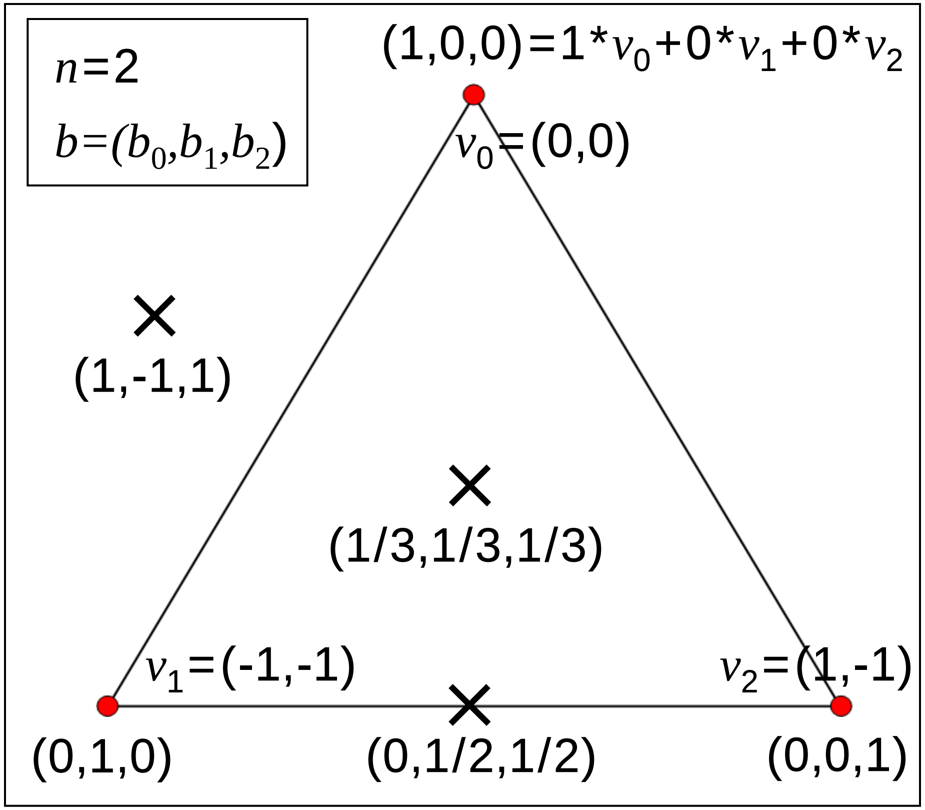
Cartesian-to-Barycentric coordinate transformation matrix:

$$A = \begin{bmatrix} v_{1_x} - v_{0_x} & v_{2_x} - v_{0_x} \\ v_{1_y} - v_{0_y} & v_{2_y} - v_{0_y} \end{bmatrix}$$

Barycentric coordinate of (x, y):

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} x - v_{0_x} \\ y - v_{0_y} \end{bmatrix}$$

$$b_0 = 1 - b_1 - b_2$$



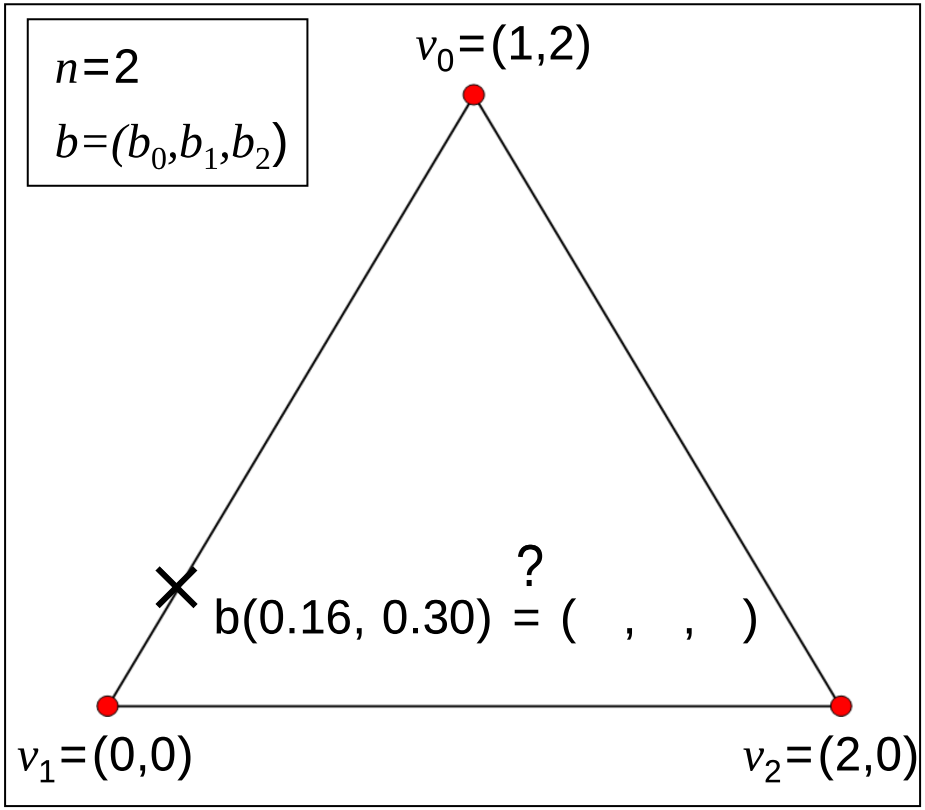
Simplex Polynomials

Exercise 1: Barycentric coordinate transformation

$$A = \begin{bmatrix} v_{1_x} - v_{0_x} & v_{2_x} - v_{0_x} \\ v_{1_y} - v_{0_y} & v_{2_y} - v_{0_y} \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} x - v_{0_x} \\ y - v_{0_y} \end{bmatrix}$$

$$b_0 = 1 - b_1 - b_2$$



Simplex Polynomials

Exercise 1: Barycentric coordinate transformation

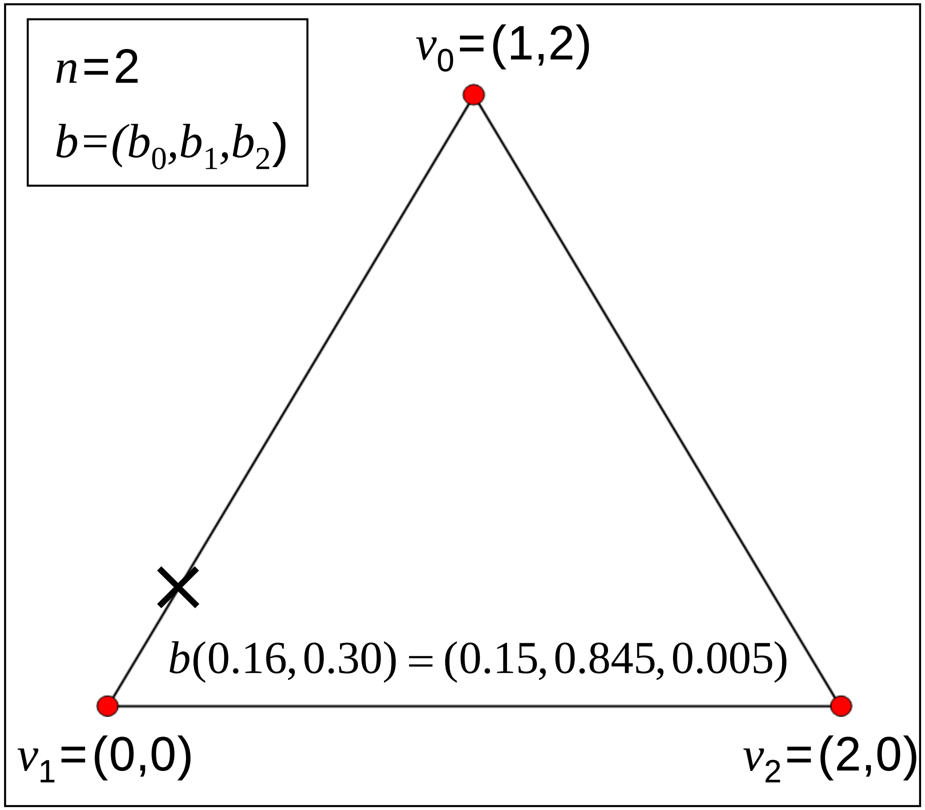
$$A = \begin{bmatrix} -1 & 1 \\ -2 & -2 \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -2 & -2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0.16-1 \\ 0.30-2 \end{bmatrix}$$

$$= \begin{bmatrix} -1/2 & -1/4 \\ 1/2 & -1/4 \end{bmatrix} \cdot \begin{bmatrix} -0.84 \\ -1.7 \end{bmatrix}$$

$$= \begin{bmatrix} 0.845 \\ 0.005 \end{bmatrix}$$

$$b_0 = 1-(0.845+0.005) = 0.15$$



Simplex Polynomials

Polynomials in Barycentric coordinates

We are interested in polynomials in terms of barycentric coordinates:

$$p(b_0, b_1, \ldots, b_n) = \left(b_0 + b_1 + \cdots + b_n\right)^d$$

Multinomial Theorem for general polynomials:

The Multinomial Theorem

$$(b_0 + b_1 + \cdots + b_n)^d = \sum_{\kappa_0 + \kappa_1 + \cdots + \kappa_n = d} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}$$

with $d \geq 0, \quad (\kappa_0, \kappa_1, \ldots, \kappa_n) \in N$

Simplex Polynomials

Polynomials in Barycentric coordinates: Multi-index notation

We have a summation of basis functions:

$$\left(b_0 + b_1 + \cdots + b_n\right)^d = \sum_{\kappa_0+\kappa_1+\cdots+\kappa_n=d} \underbrace{\frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}}$$

Single basis function: $\dfrac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} = B_{\kappa_0,\kappa_1,\ldots,\kappa_n}(b_0, b_1, \ldots, b_n)$

We now introduce the **multi-index** $\kappa = (\kappa_0, \kappa_1, \ldots, \kappa_n)$.

The multi-index has the following properties:

$$|\kappa| = \kappa_0 + \kappa_1 + \cdots + \kappa_n$$

$$\kappa! = \kappa_0!\kappa_1!\cdots\kappa_n!$$

Simplex Polynomials

B-form basis functions

The multi-index $\kappa = (\kappa_0, \kappa_1, \dots, \kappa_n)$ allows us to simplify and generalize the notation for the basis polynomials.

Using the multi-index, the basis functions can be greatly simplified:

$$\frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} \cdot b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} = \frac{d!}{\kappa!} \cdot b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}$$

$$= \frac{d!}{\kappa!} \cdot b^{\kappa}$$

$$= B_{\kappa}^{d}(b)$$

$\kappa! = \kappa_0!\kappa_1!\cdots\kappa_n!$

$b^{\kappa} = b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}$

Definition…

Simplex Polynomials

Summation of multivariate spline basis functions

$$\left(b_0 + b_1 + \cdots + b_n\right)^d = \sum_{\kappa_0+\kappa_1+\cdots+\kappa_n=d} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}$$

$$= \sum_{|\kappa|=d} \frac{d!}{\kappa!} \cdot b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}$$

$$= \sum_{|\kappa|=d} \frac{d!}{\kappa!} \cdot b^{\kappa}$$

$$= \sum_{|\kappa|=d} B_{\kappa}^{d}(b)$$

$$= 1$$

$$\kappa! = \kappa_0!\kappa_1!\cdots\kappa_n!$$
$$|\kappa| = \kappa_0 + \kappa_1 + \cdots + \kappa_n$$

$$b^{\kappa} = b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n}$$

This is what is called a **Stable Local Basis.**

- *Stable* because summation of all terms = 1.
- *Local* because basis functions are only locally active and 0 everywhere else.

Simplex Polynomials

Example 6.3: Expanding a polynomial using the multinomial theorem

$$(b_0+b_1+b_2)^3=\sum_{|\kappa|=3}B_\kappa^3(b)=\sum_{\kappa_0+\kappa_1+\kappa_2=3}\frac{3!}{\kappa_0!\kappa_1!\kappa_2!}b_0^{\kappa_0}b_1^{\kappa_1}b_2^{\kappa_2}$$

Step 1: write out the **valid permutations** of $\kappa_0,\kappa_1,\kappa_2$ for d=3:

$$(\kappa_0,\kappa_1,\kappa_2)\in\left\{\begin{array}{l}(3,0,0),\\(2,1,0),(2,0,1)\\(1,2,0),(1,1,1),(1,0,2)\\(0,3,0),(0,2,1),(0,1,2),(0,0,3)\end{array}\right\}$$

For a valid permutation we **must** have:
$|\kappa|=\kappa_0+\kappa_1+\kappa_2=3$

Simplex Polynomials

Example 6.3: Expanding a polynomial using the B-form

Step 2: expand the summation using the values found for $\kappa_0, \kappa_1, \kappa_2$

$$
\begin{aligned}
(b_0+b_1+b_2)^3 &= \frac{3!}{3!0!0!}b_0^3b_1^0b_2^0+\frac{3!}{2!1!0!}b_0^2b_1^1b_2^0+\frac{3!}{2!0!1!}b_0^2b_1^0b_2^1+ \\
&\quad \frac{3!}{1!2!0!}b_0^1b_1^2b_2^0+\frac{3!}{1!1!1!}b_0^1b_1^1b_2^1+\frac{3!}{1!0!2!}b_0^1b_1^0b_2^2+ \\
&\quad \frac{3!}{0!3!0!}b_0^0b_1^3b_2^0+\frac{3!}{0!2!1!}b_0^0b_1^2b_2^1+\frac{3!}{0!1!2!}b_0^0b_1^1b_2^2+\frac{3!}{0!0!3!}b_0^0b_1^0b_2^3 \\
&= b_0^3+3b_0^2b_1^1+3b_0^2b_2^1+3b_0^1b_1^2+6b_0^1b_1^1b_2^1+3b_0^1b_2^2+ \\
&\quad b_1^3+3b_1^2b_2^1+3b_1^1b_2^2+b_2^3
\end{aligned}
$$

Simplex Polynomials

Multinomial Theorem for polynomials in barycentric coordinates

$$(b_0 + b_1 + \cdots + b_n)^d = \sum_{|\kappa|=d} \frac{d!}{\kappa!} \cdot b^\kappa = \sum_{|\kappa|=d} B_\kappa^d(b) = 1$$

WHY ALL THE TROUBLE???

De Boor's Theorem (1987)

Any polynomial $p(x)$ of degree d can be written in the **B-form** as follows:

$$p(x) = \sum_{|\kappa|=d} c_\kappa^{t_j} B_\kappa^d(b_{t_j}(x))$$

With $c_\kappa^{t_j}$ the polynomial coefficients, or **B-coefficients**, and with $b = (b_0, b_1, \dots, b_n)$ the barycentric coordinates of x with respect to an n-simplex t_j.

Simplex Polynomials

The Vector form of the B-form

De Boor's formulation of the B-form is:

$$p(x) = \sum_{|\kappa|=d} c_{\kappa}^{t_j} B_{\kappa}^{d}(b_{t_j}(x)) \qquad \forall x \in t_j$$

This formulation can be changed in a more practical **vector notation(*)**:

$$\boxed{p(x) = B^{d}(b_{t_j}(x)) \cdot c^{t_j} \qquad \forall x \in t_j}$$

By definition we also have:

$$\boxed{p(x) = 0 \quad \forall x_i \notin t_j}$$

(*)C.C. de Visser, Q.P. Chu, and J.A. Mulder, A New Approach for Linear Regression with Multivariate Splines, Automatica, 2009.

Simplex Polynomials

The Vector form of the B-form

NOTE: Many errors in calculations with splines are caused by incorrectly sorting the basis polynomials and B-coefficients!

We have the vector form of the B-form as follows:

$$\boxed{p(x) = B^d(b_{t_j}(x)) \cdot c^{t_j}}$$

EXTREMELY IMPORTANT: with $B^d(b)$ the **SORTED** vector of basis polynomials in terms of the barycentric coordinate b:

$$\boxed{B^d(b_{t_j}(x)) = \begin{bmatrix} B^d_{d,0,0}(b_{t_j}(x)) & B^d_{d-1,1,0}(b_{t_j}(x)) & \cdots & B^d_{0,1,d-1}(b_{t_j}(x)) & B^d_{0,0,d}(b_{t_j}(x)) \end{bmatrix}}$$

and with c^{t_j} the vector of **CORRESPONDINGLY SORTED** B-coefficients for a simplex t_j which are sorted as follows:

$$\boxed{c^{t_j} = \begin{bmatrix} c^{t_j}_{d,0,0} & c^{t_j}_{d-1,1,0} & \cdots & c^{t_j}_{0,1,d-1} & c^{t_j}_{0,0,d} \end{bmatrix}^T}$$

Simplex Polynomials

Some interesting properties of the B-form:

Analytical integral of a B-form over a simplex (T) equals a constant times the sum of all B-coefficients:

$$\int_t B^d(b_{t_j}(x)) \cdot c^t db = \frac{\text{vol}(t) \cdot n! d!}{(d+n)!} \sum_{|\kappa|=d} c_\kappa^t$$

Directional derivatives can be expressed in the **original** B-coefficient vector(!):

$$D_u^m p(b_{t_j}(x)) = \frac{d!}{(d-m)!} \mathrm{B}^{d-m}(b_{t_j}(x)) \cdot \mathrm{P}^{d,d-m}(a_{t_j}(u)) \cdot \mathrm{c}$$

original B-coefficients

“de Casteljau”(*) matrix

(*)C.C. de Visser, Q.P. Chu, and J.A. Mulder, Differential constraints for bounded recursive identification with multivariate splines. *Automatica*, 2011.

Simplex Polynomials

Exercise 2: Simplex polynomial evaluation

Evaluate the quadratic B-form polynomial

$$p(b) = \sum_{|\kappa|=d} c_\kappa^{t_j} B_\kappa^2(b_{t_j}) = B^2(b_{t_j}) \cdot c^{t_j}$$

at the barycentric coordinate

$$b = (0.5,\ 0.25,\ 0.25)$$

when it is given that

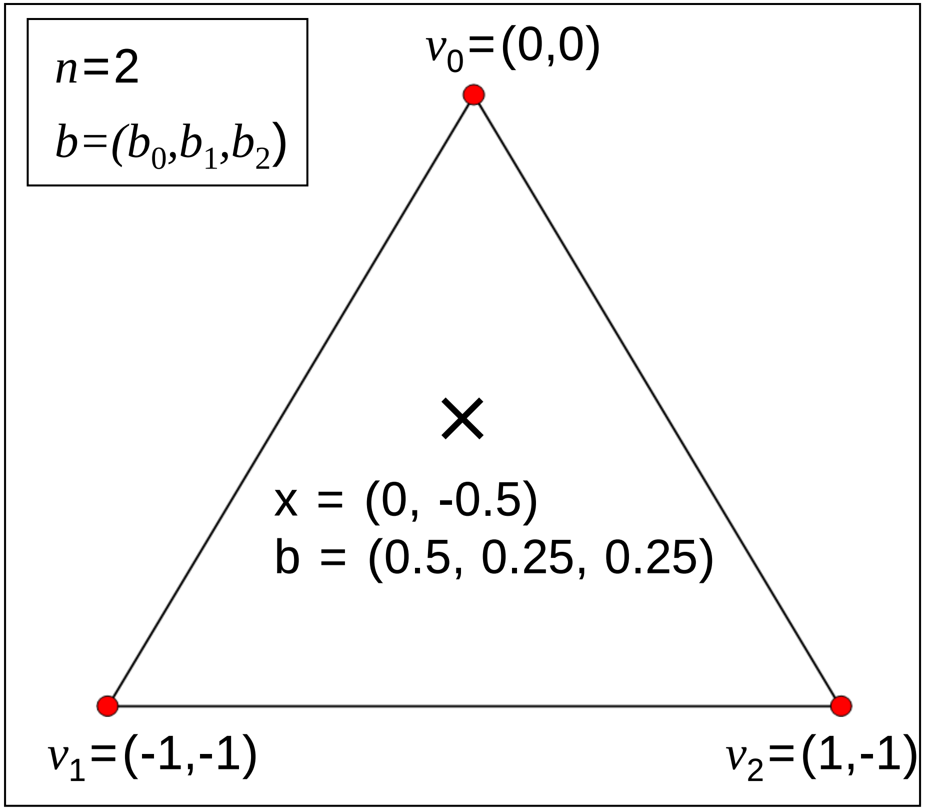
$$c_{2,0,0} = 2$$

$$c_{1,1,0} = -4 \qquad c_{1,0,1} = 4$$

$$c_{0,2,0} = -4 \qquad c_{0,1,1} = 3 \qquad c_{0,0,2} = 2$$

answer:

$$p(0.5,\ 0.25,\ 0.25) =$$



Simplex Polynomials

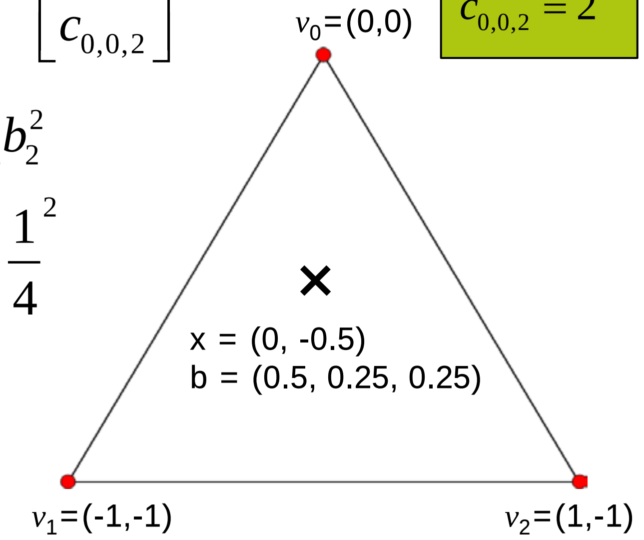
Exercise 2: Simplex polynomial evaluation $p(b) = B^2(b_{t_j}) \cdot c^{t_j}$

$$p(b) = \begin{bmatrix} b_0^2 b_1^0 b_2^0 & 2b_0^1 b_1^1 b_2^0 & 2b_0^1 b_1^0 b_2^1 & b_0^0 b_1^2 b_2^0 & 2b_0^0 b_1^1 b_2^1 & b_0^0 b_1^0 b_2^2 \end{bmatrix} \begin{bmatrix} c_{2,0,0} \\ c_{1,1,0} \\ c_{1,0,1} \\ c_{0,2,0} \\ c_{0,1,1} \\ c_{0,0,2} \end{bmatrix}$$

$c_{2,0,0} = 2$
$c_{1,1,0} = -4$
$c_{1,0,1} = 4$
$c_{0,2,0} = -4$
$c_{0,1,1} = 3$
$c_{0,0,2} = 2$

$$= c_{2,0,0} b_0^2 + c_{1,1,0} 2b_0^1 b_1^1 + c_{1,0,1} 2b_0^1 b_2^1 + c_{0,2,0} b_1^2 + c_{0,1,1} 2b_1^1 b_2^1 + c_{0,0,2} b_2^2$$

$$= 2 \cdot {\frac{1}{2}}^2 + -4 \cdot 2 \cdot \frac{1}{2} \cdot \frac{1}{4} + 4 \cdot 2 \cdot \frac{1}{2} \cdot \frac{1}{4} + -4 \cdot {\frac{1}{4}}^2 + 3 \cdot 2 \cdot \frac{1}{4} \cdot \frac{1}{4} + 2 \cdot {\frac{1}{4}}^2$$

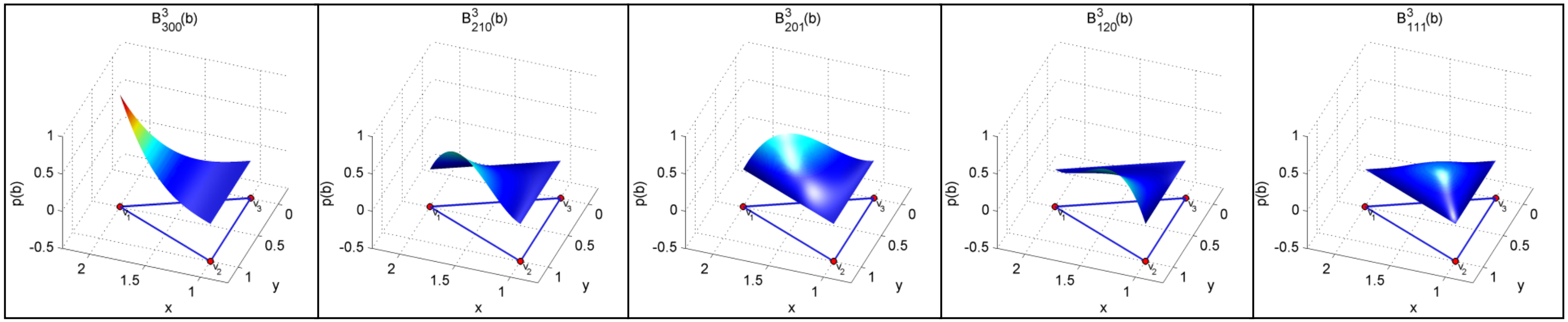


answer:

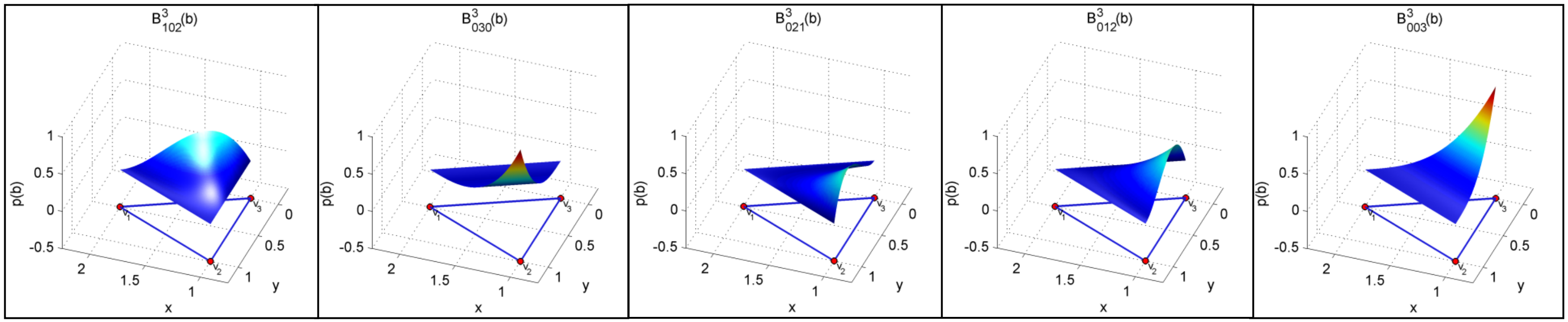
$$p(0.5,\ 0.25,\ 0.25) = 0.75$$

Simplex Polynomials

Individual basis functions of the B-form $B^3(b) \cdot c$ with $c = 1$



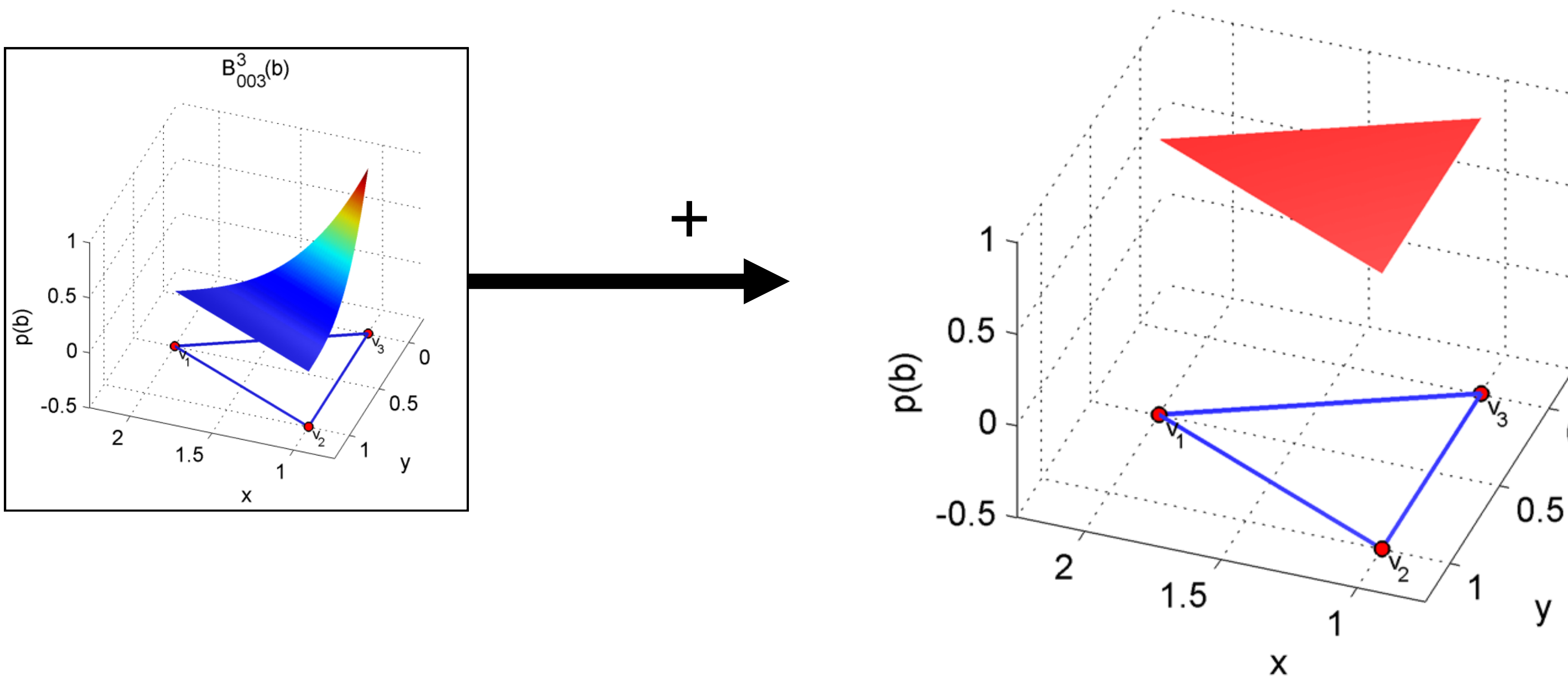
$b_0^3 b_1^0 b_2^0$ $\quad 3b_0^2 b_1^1 b_2^0$ $\quad 3b_0^2 b_1^0 b_2^1$ $\quad 3b_0^1 b_1^2 b_2^0$ $\quad 6b_0^1 b_1^1 b_2^1$



$3b_0^1 b_1^0 b_2^2$ $\quad b_0^0 b_1^3 b_2^0$ $\quad 3b_0^0 b_1^2 b_2^1$ $\quad 3b_0^0 b_1^1 b_2^2$ $\quad b_0^0 b_1^0 b_2^3$

Simplex Polynomials

Individual basis functions of the B-form $B^3(b) \cdot c$ with $c = 1$



$$p(b) = b_0^3 b_1^0 b_2^0$$

Simplex Polynomials

B-coefficients control the exact shape of simplex polynomials...

B-coefficients

The B-coefficients **locally control** the shape of the simplex polynomial.

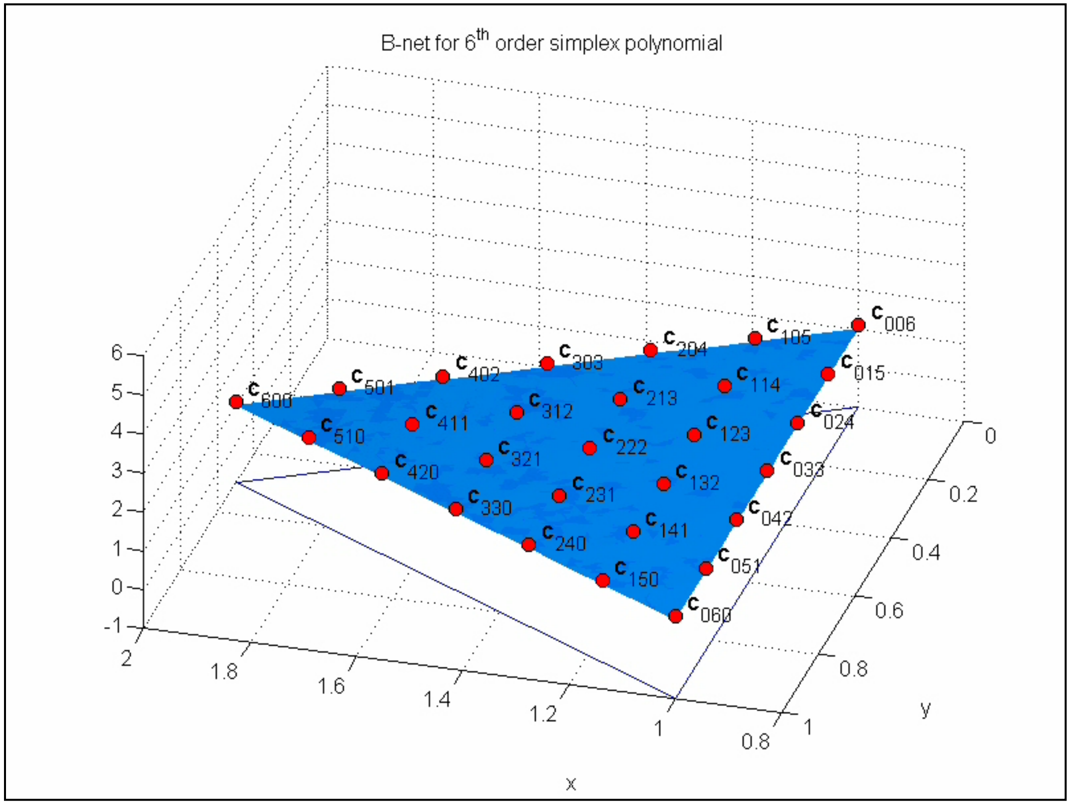
The B-coefficients have a **unique spatial location** within a simplex.

The structure of the B-coefficients within a simplex is called the **B-net**.

Location of B-coefficients

The location of any B-coefficient in barycentric coordinates is:

$$b(c_k) = \frac{\kappa}{d}$$



6^{th} degree simplex polynomial ($p(b) = B^6(b) \cdot c$) with 28 B-coefficients

Simplex Polynomials

B-net structure

Total number of B-coefficients

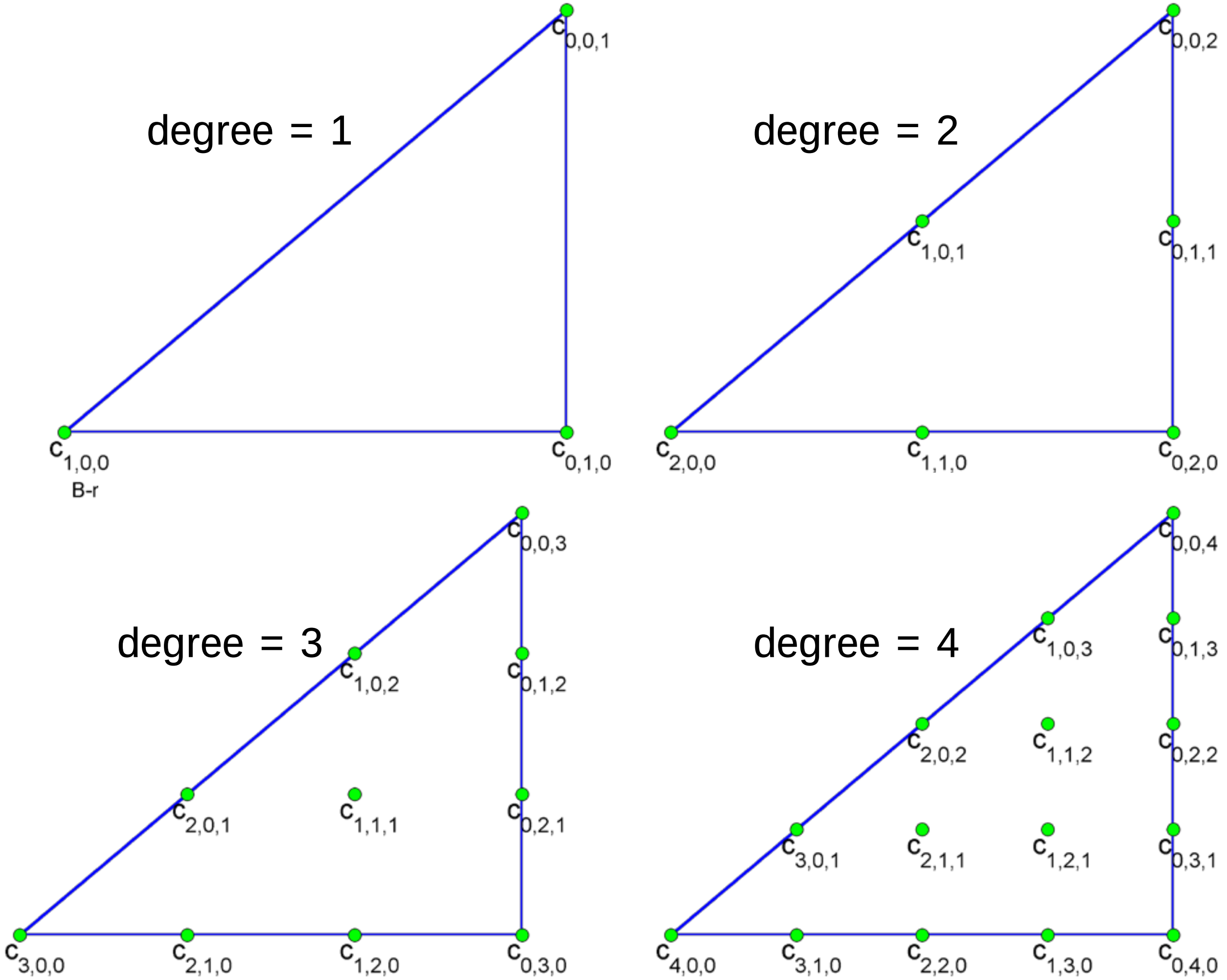
The total number of B-coefficients for a given degree *d* and dimension *n* is:

$$\hat{d} = \binom{d+n}{n} = \frac{(d+n)!}{n!d!}$$

Location of B-coefficients

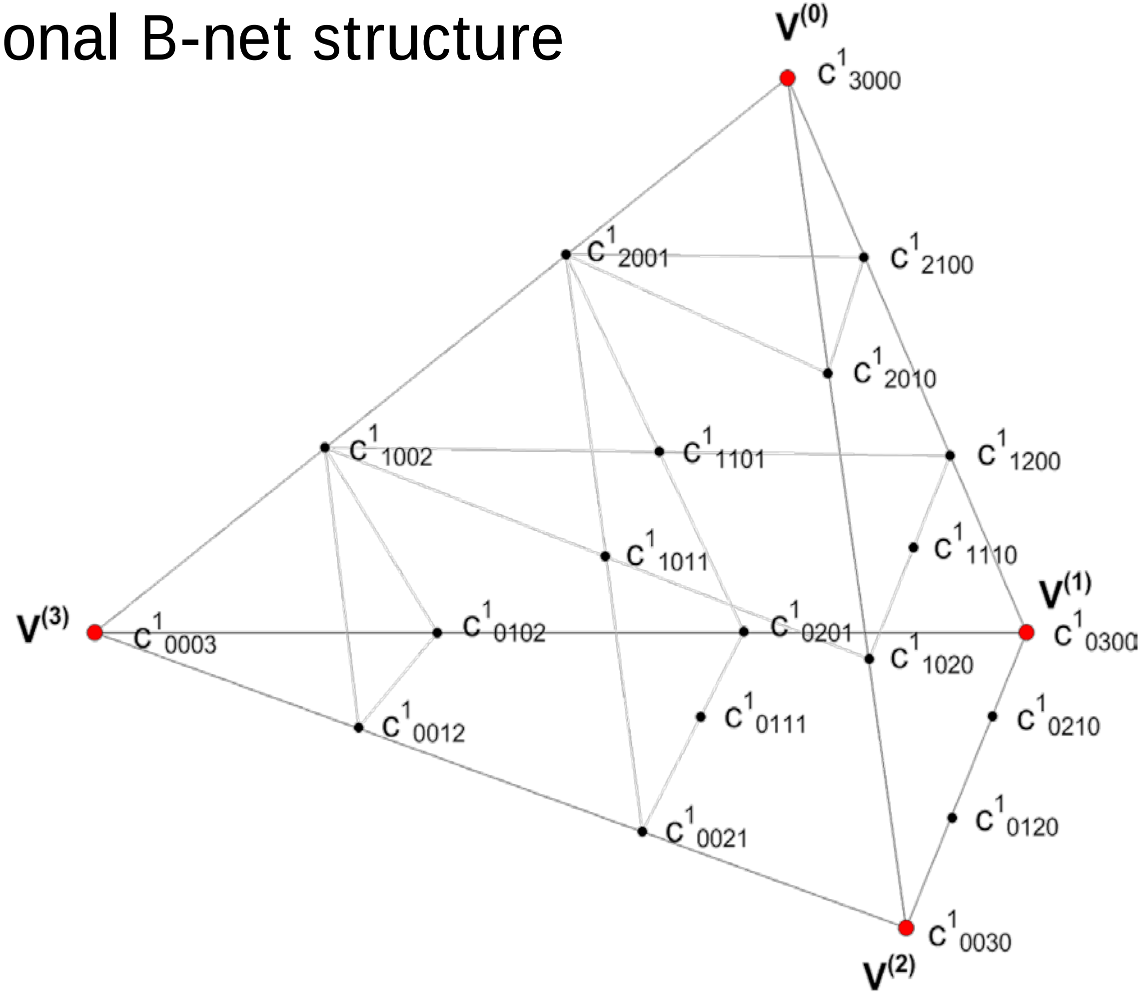
The location of any B-coefficient in barycentric coordinates is:

$$b(c_k) = \frac{\kappa}{d}$$



Simplex Polynomials

Higher dimensional B-net structure



Simplex Polynomials

B-net Orientation Rule

- The B-net has a very specific orientation within its parent simplex.
- This orientation is closely related to the earlier mentioned **sorting** of terms.
- This orientation will prove to be **essential** when defining **inter-simplex continuity**.

The B-net orientation rule:
The B-coefficient with the **highest multi-index** value should be located at the vertex with the **lowest index**, the B-coefficient with the **second highest multi-index** value should be located at the vertex with the **second lowest index** and so on.

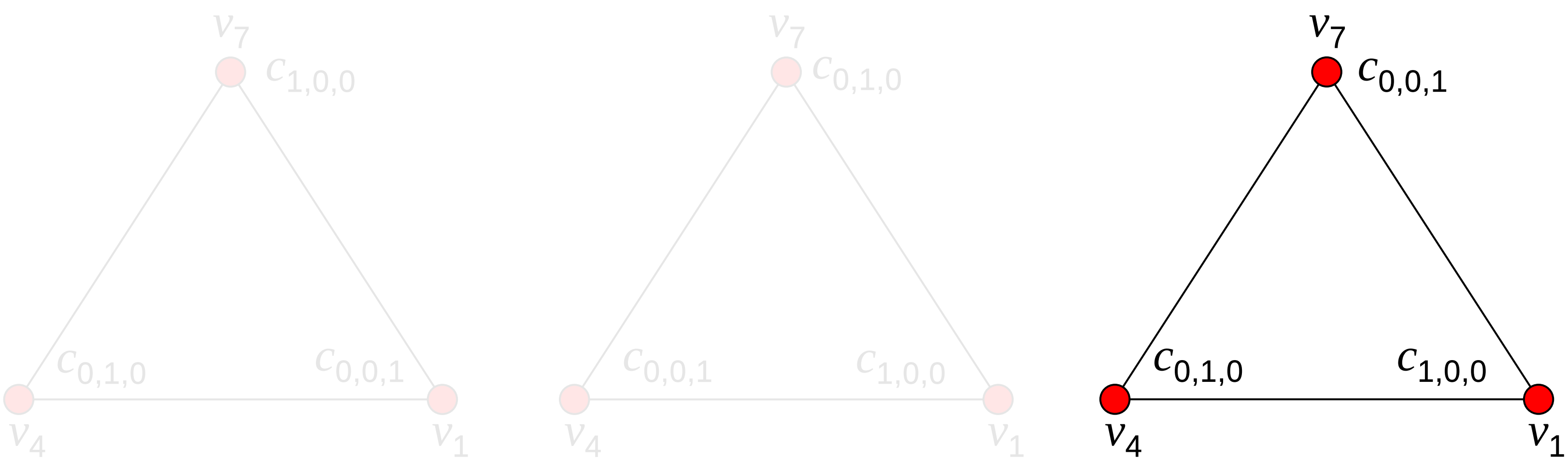
NOTE: Many errors in calculations with splines are caused by incorrectly applying the B-net orientation rule!

Simplex Polynomials

B-net Orientation Rule

The B-coefficient with the **highest multi-index** value should be located at the vertex with the **lowest index**, the B-coefficient with the **second highest multi-index** value should be located at the vertex with the **second lowest index** and so on.

Example:



Simplex Polynomials

Exercise: B-net Orientation Rule

Plot all B-nets for a quadratic bivariate (2D) spline function:

$$p(b_t(x,y)) = \sum_{|\kappa|=2} c_\kappa^{t_j} B_\kappa^2(b_t(x,y))$$

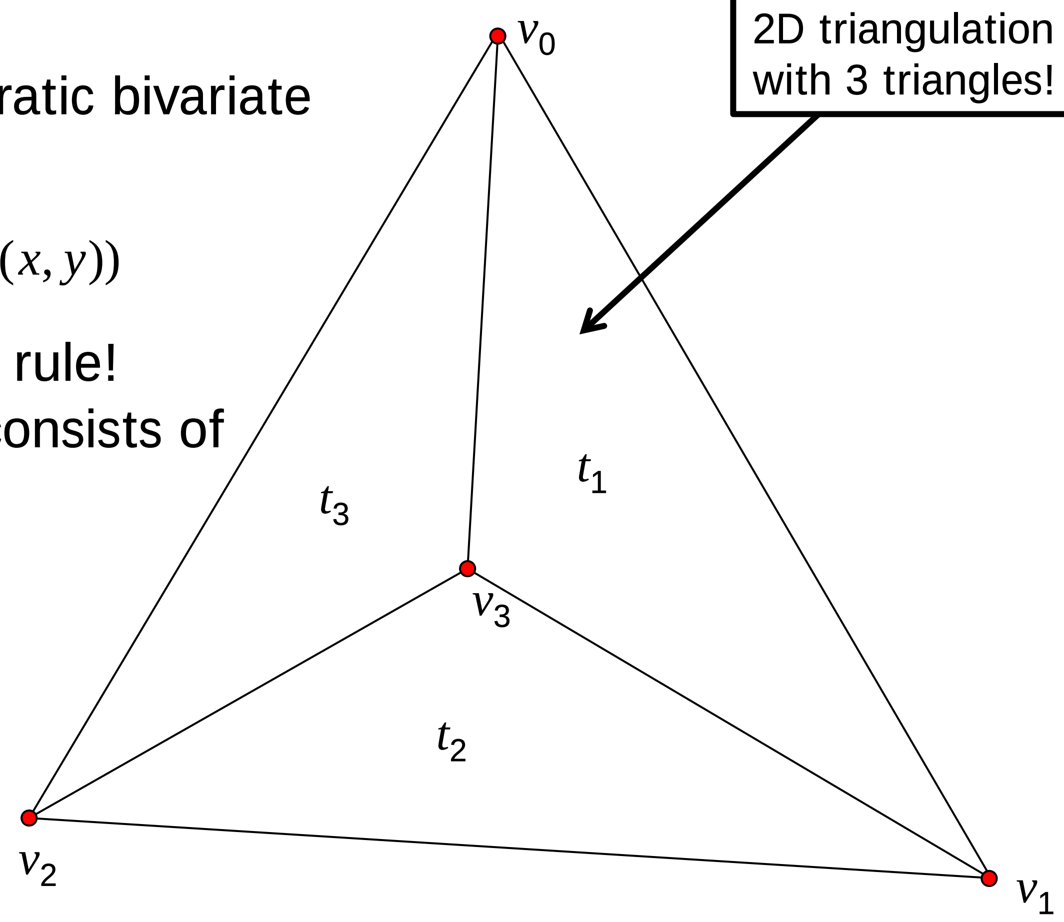
use the B-net orientation rule!

Note: the triangulation consists of 3 triangles!

B-net orientation rule:

The B-coefficient with the **highest multi-index** value should be located at the vertex with the **lowest index**, the B-coefficient with the **second highest multi-index** value should be located at the vertex with the **second lowest index** and so on.

2D triangulation with 3 triangles!



Simplex Polynomials

Summary Simplex Polynomials

- Simplices have a natural local coordinate system: **barycentric coordinates**.
- Simplex polynomials are **ordinary polynomials** in barycentric coordinates.
- Simplex polynomials can be written in the **B-form**.
- Simplex polynomials have a **stable local basis**; all terms sum up to 1, and the polynomials are only defined on their parent simplex.
- The exact shape of simplex polynomials is determined by the **B-coefficients**.
- The B-coefficients have a **unique spatial location** within a simplex which is determined by the **B-net orientation rule**.

Simplex Polynomials

Summary Mathematics of Simplex Polynomials

The B-form: $p(b) = \sum_{|\kappa|=d} \left(c_\kappa \cdot B_\kappa^d \right) = \sum_{|\kappa|=d} \left(c_\kappa \cdot \frac{d!}{\kappa!} b^\kappa \right)$

The **sorted** vector B-form: $p(b) = B^d(b) \cdot c^{t_j}$

The **sorted** vector B-form with implicit barycentric coordinate transformation:

$$p(x) = B^d(b_{t_j}(x)) \cdot c^{t_j} \quad \forall x \in t_j$$

Example: $b = (b_0, b_1, b_2)$, degree = 3:

$$p(b) = \sum_{|\kappa|=3} \left(c_\kappa \cdot B_\kappa^3(b) \right) = B^3(b) \cdot c^{t_j}$$

$$= \sum_{|\kappa|=3} \left(c_\kappa \cdot \frac{d!}{\kappa!} b_0^{\kappa_0} b_1^{\kappa_1} b_2^{\kappa_2} \right)$$

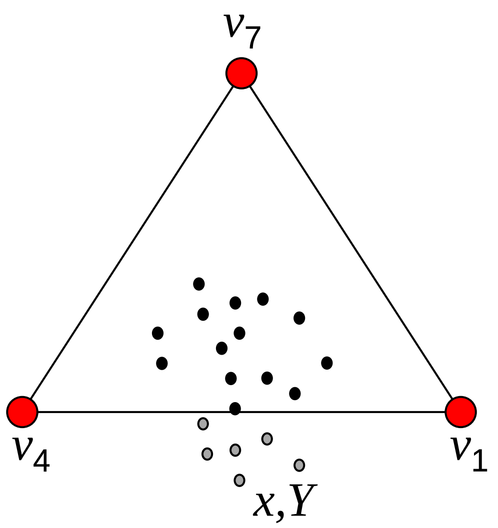
with

$d!/\kappa!$	κ_0	κ_1	κ_2
1	3	0	0
3	2	1	0
3	2	0	1
3	1	2	0
6	1	1	1
3	1	0	2
1	0	3	0
3	0	2	1
3	0	1	2
1	0	0	3

OLS with simplex polynomials

The 8 steps to estimate a simplex polynomial

We can use our knowledge of linear regression to create an OLS estimator for the B-coefficients of a simplex polynomial.



Step 1: We need a set of measurements (x, Y):

$$x \in \mathbb{R}^{m \times n},\ Y = \mathbb{R}^{m \times 1}$$

Step 2: We then define a set of $n+1$ vertices V:

$$V \in \mathbb{R}^{n+1 \times n}$$

Step 3: We define the simplex t as the convex hull of V :

$$t = \langle V \rangle$$

OLS with simplex polynomials

The 8 steps to estimate a simplex polynomial

Step 4: We do a data point membership search: which data points x are inside our simplex t? For this we can use the Matlab function `tsearchn`:

```
[IMap, BaryC] = tsearchn(X,TRI,XI);
```

Matlab tip!

Step 5: We transform all datapoints that are inside t into barycentric coordinates:

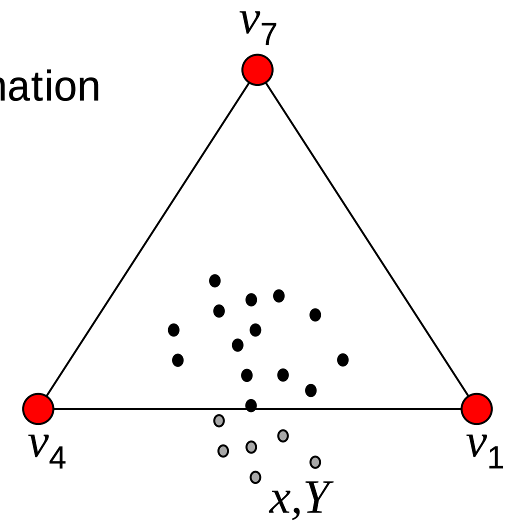
explicit transformation

$$\begin{bmatrix} b_1 & b_2 & \cdots & b_{n-1} \end{bmatrix}^T = A_{t_j}^{-1} \cdot \left(x^T(i) - v_0^T \right)$$

$$b_0 = 1 - \sum_{i=1}^{n} b_i$$

implicit transformation

$$b = b_{t_j}\left(x_i\right)$$



Again, Matlab can help us out here: the function `tsearchn` also returns the barcycentric coordinates (`BaryC`)!

OLS with simplex polynomials

The 8 steps to estimate a simplex polynomial

Step 6: We formulate a simplex polynomial model structure using the vector B-form. For a single simplex, this only requires a choice for d:

$$p(x) = B^d(b_{t_j}(x)) \cdot c^{t_j}$$

Step 7: Create the **sorted** B-form regression matrix B for M measurements:

$$B = \begin{bmatrix} B^d_{d,0,0}(b(1)) & B^d_{d-1,1,0}(b(1)) & \cdots & B^d_{0,1,d-1}(b(1)) & B^d_{0,0,d}(b(1)) \\ B^d_{d,0,0}(b(2)) & B^d_{d-1,1,0}(b(2)) & \cdots & B^d_{0,1,d-1}(b(2)) & B^d_{0,0,d}(b(2)) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B^d_{d,0,0}(b(M)) & B^d_{d-1,1,0}(b(M)) & \cdots & B^d_{0,1,d-1}(b(M)) & B^d_{0,0,d}(b(M)) \end{bmatrix}$$

Step 8: We formulate an OLS estimator for the B-coefficients:

$$\hat{c} = \left(B^T \cdot B\right)^{-1} \cdot B^T \cdot Y$$

OLS with simplex polynomials

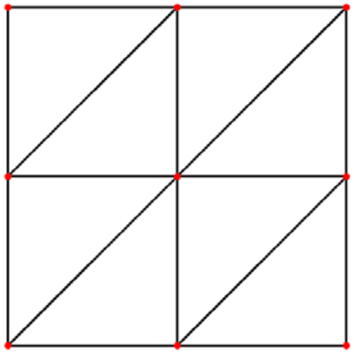
Summary Simplex Polynomials

MATLAB Demo

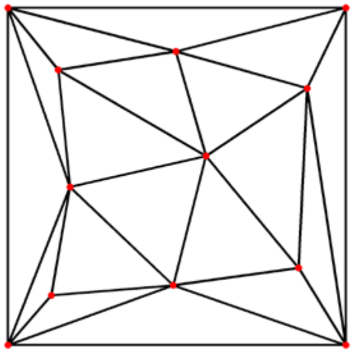
Fitting a bivariate (2-D) dataset with a single simplex polynomial.

Multivariate Simplex B-Splines

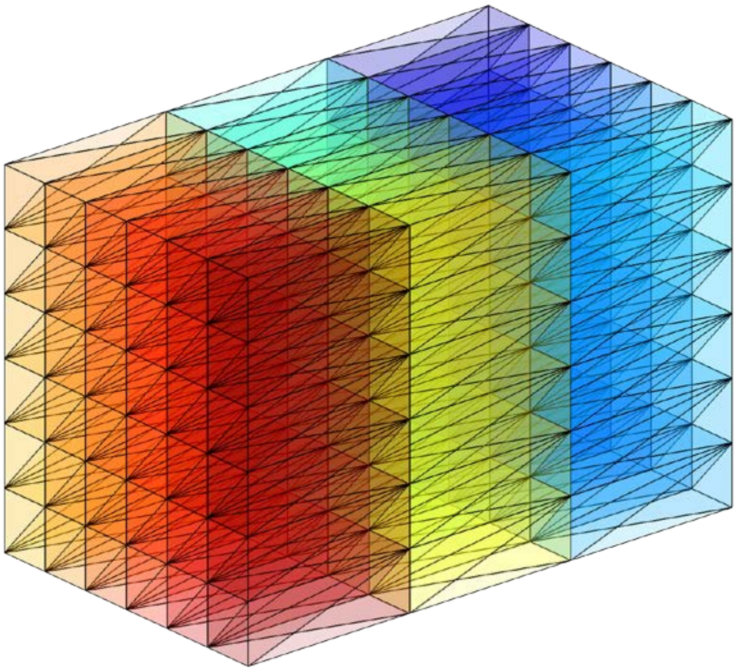
Simplex Splines are defined on Triangulations...



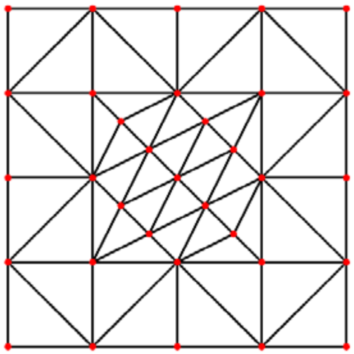
Low resolution triangulations with high degree splines?



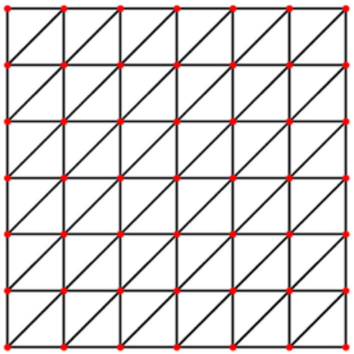
Data complexity dependent triangulations?



Higher dimensions...



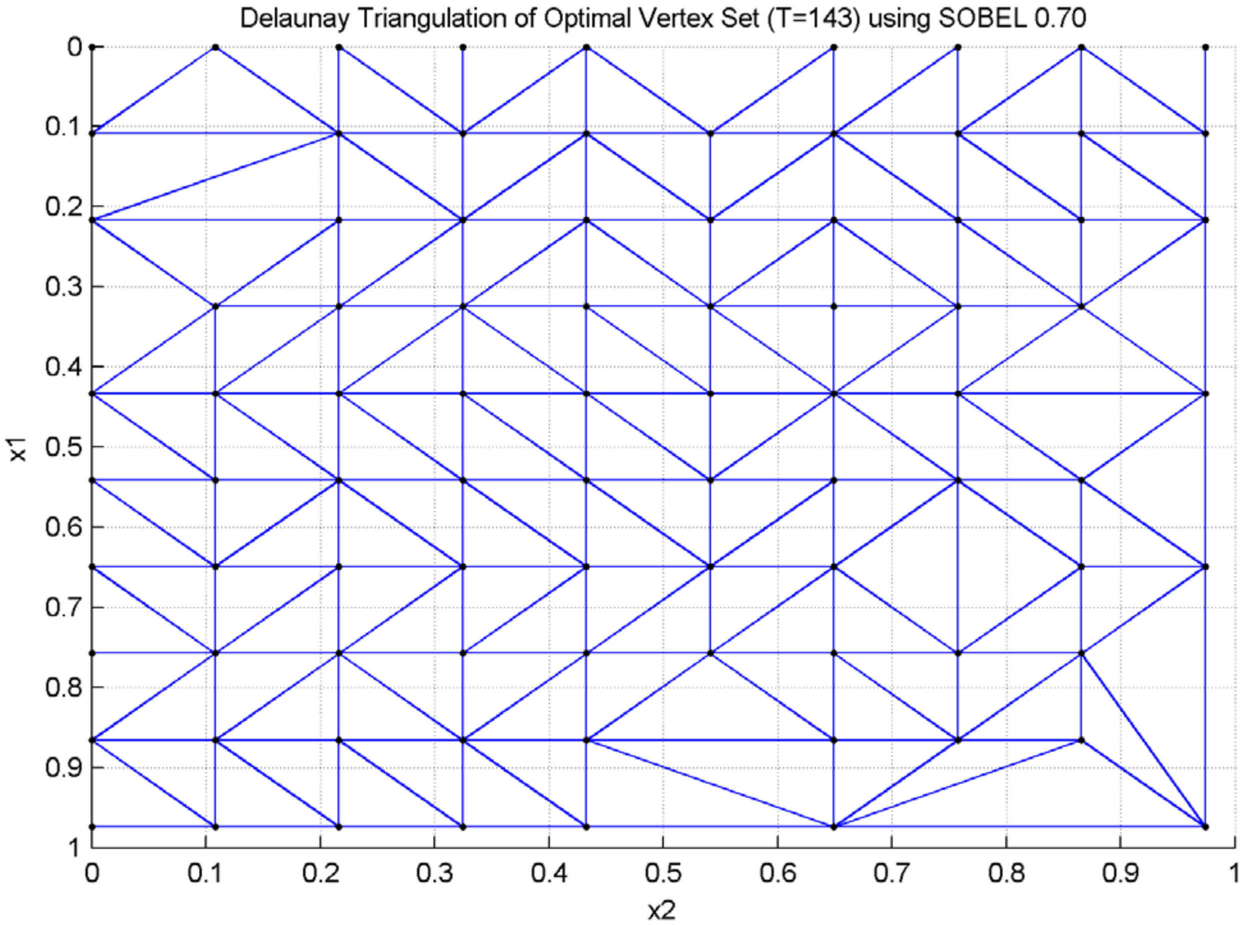
Local refinements?



High resolution triangulation with low degree splines?

Multivariate Simplex B-Splines

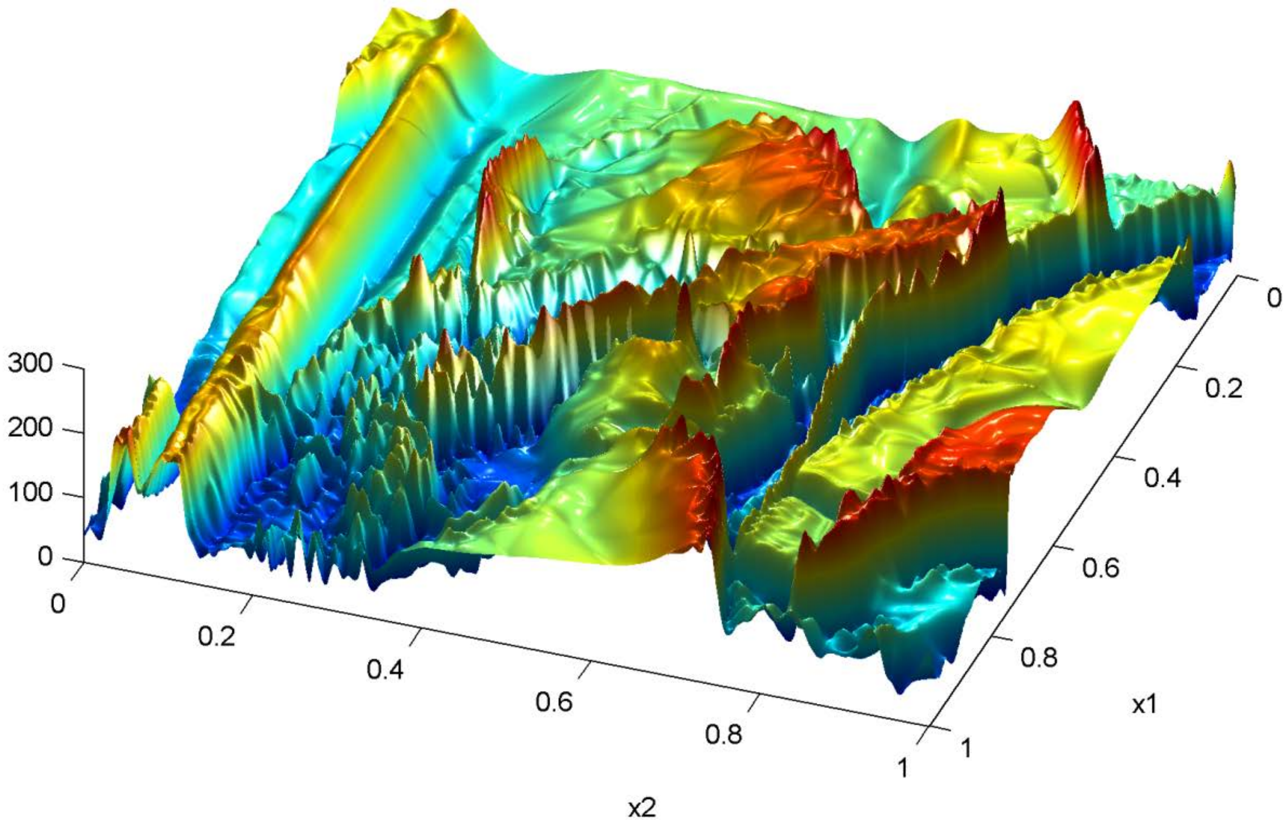
Triangulation optimization based on local complexity



Multivariate Simplex B-Splines

Triangulation optimization based on local complexity

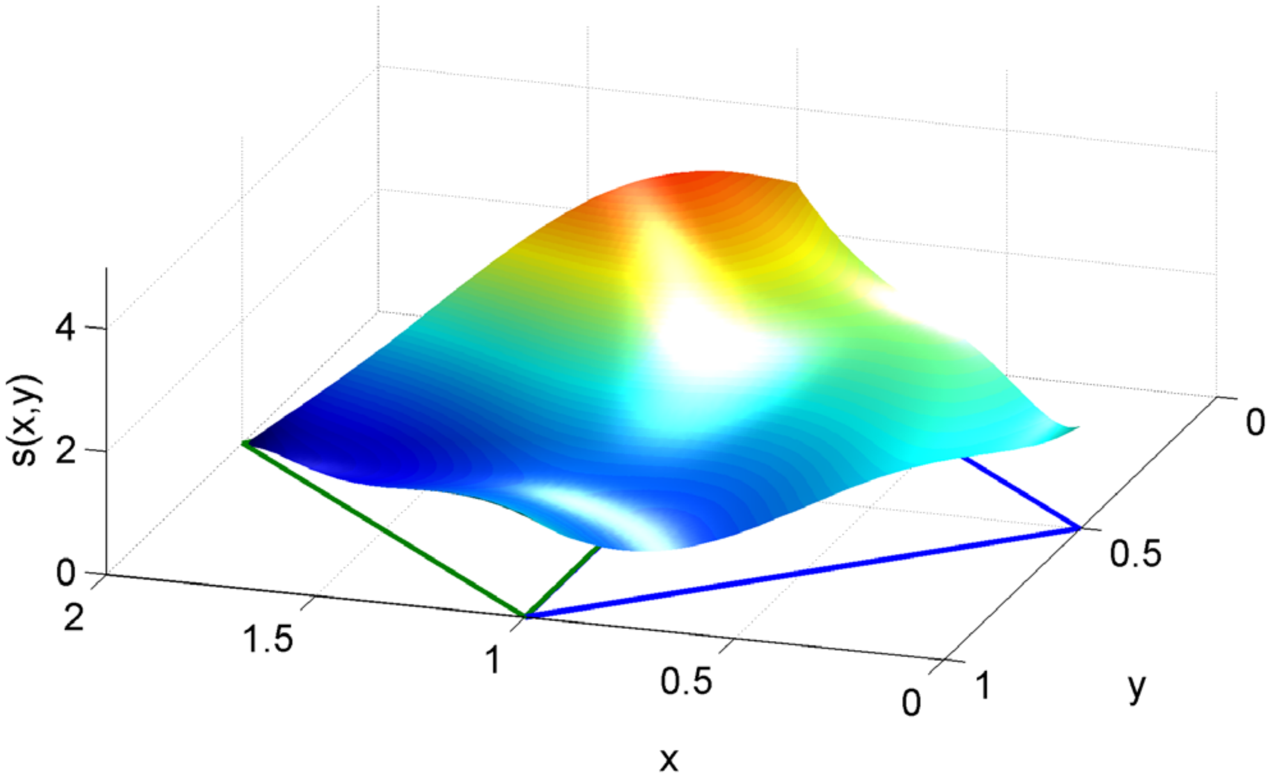
Spline of degree 3 on 5258 simplices



Continuity of Simplex B-Splines

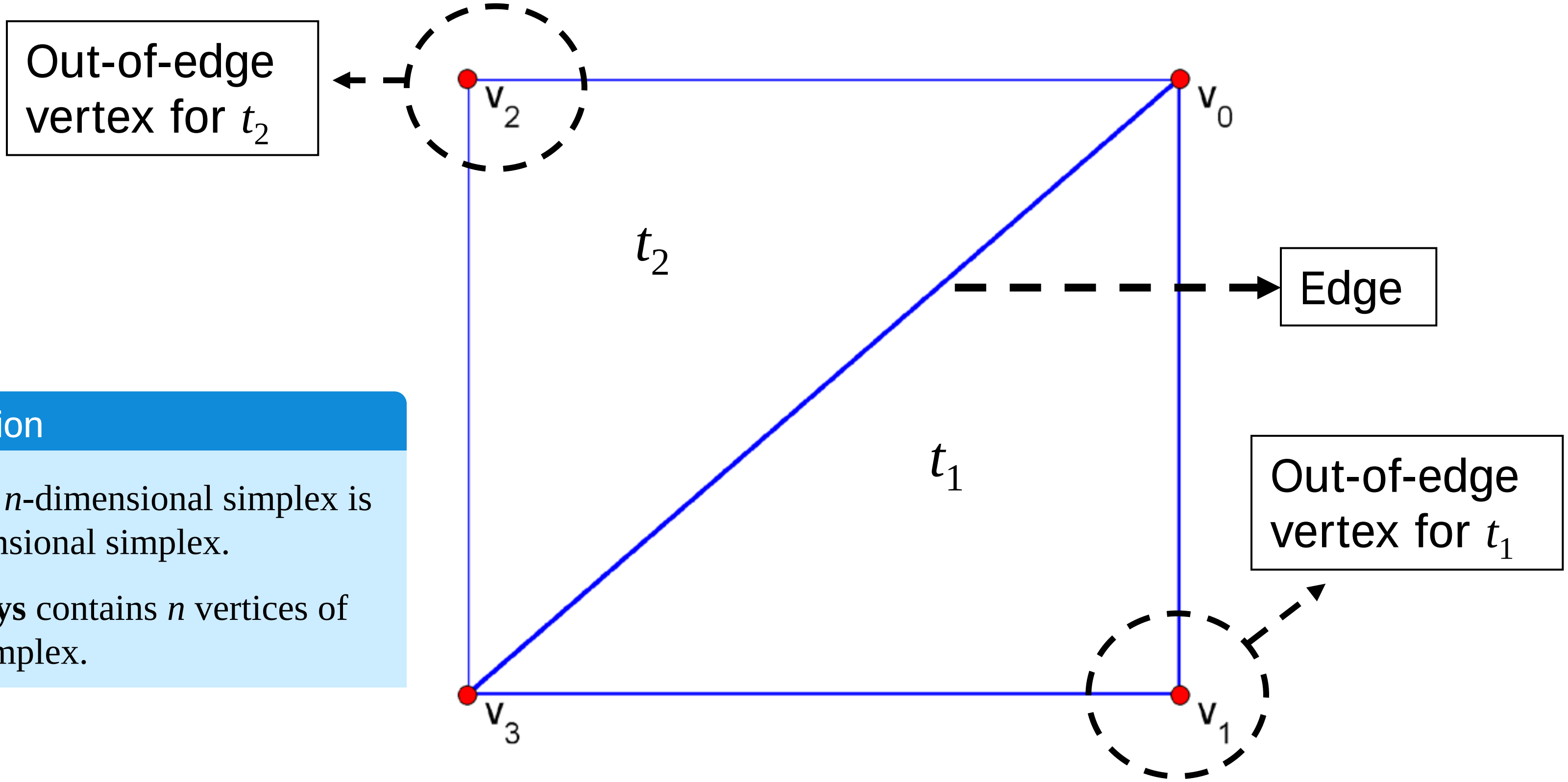
A simplex spline is a piecewise continuous function…

Two spline polynomials of degree 6 (4-th order continuity)



Continuity of Simplex B-Splines

Continuity between simplices: some definitions



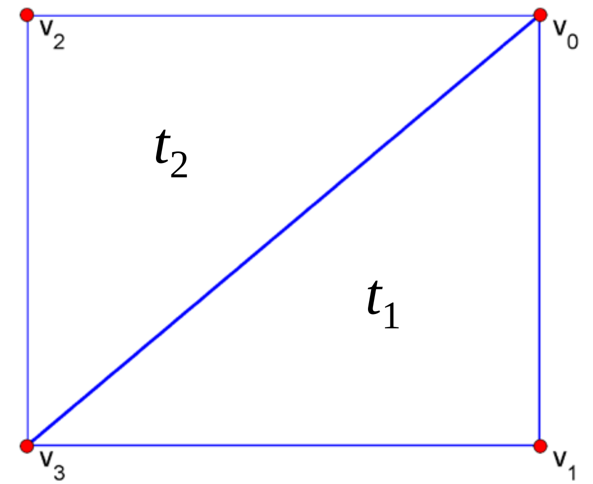
Edge: Definition

An edge of an *n*-dimensional simplex is an (*n*-1)-dimensional simplex.

An edge **always** contains *n* vertices of its parent n-simplex.

Continuity of Simplex B-Splines

Continuity between simplices



de Boor's Continuity Equations (1987)

The r^{th} order continuity between simplex polynomials defined on 2 neighboring simplices t_1 and t_2 is achieved when the following holds for the B-coefficients of these two polynomials:

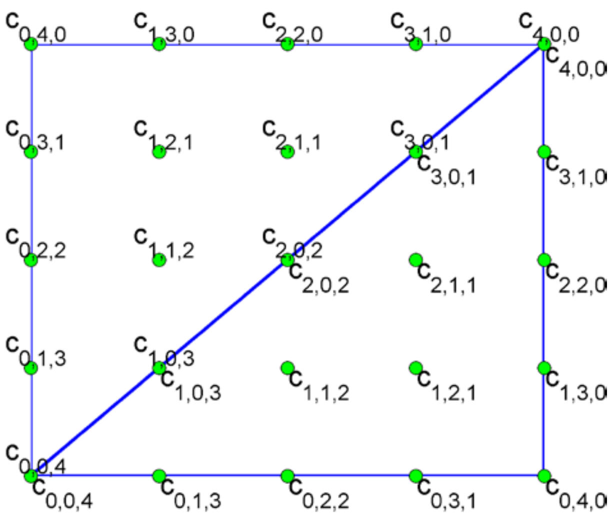
$$c^{t_2}_{\kappa_0,m,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_*), \quad 0 \le m \le r$$

with $r < d$, with γ a multi-index independent of κ and with v_* **the out-of-edge vertex** of t_2

We define:

$$\kappa_0 + m + \kappa_1 = d$$

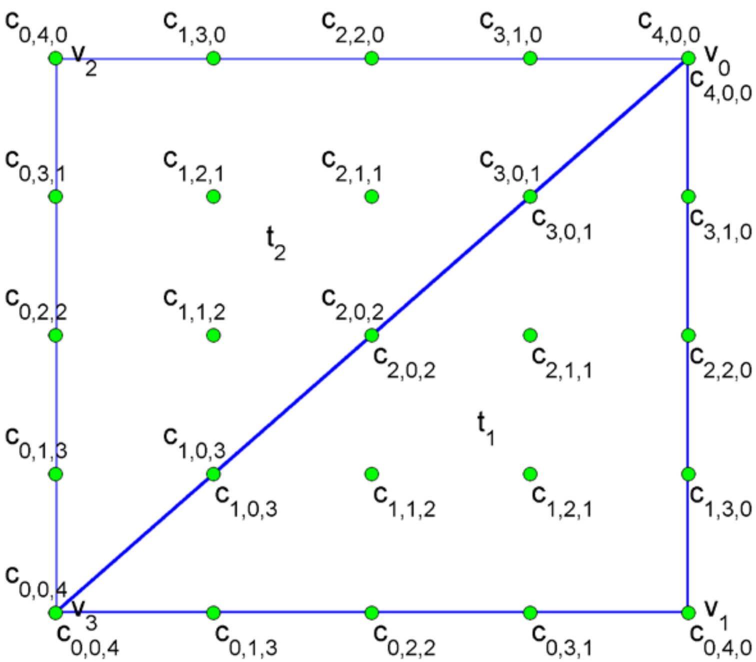
$$(\kappa_0 + 0 + \kappa_1) + (\gamma_0 + \gamma_1 + \gamma_2) = d$$



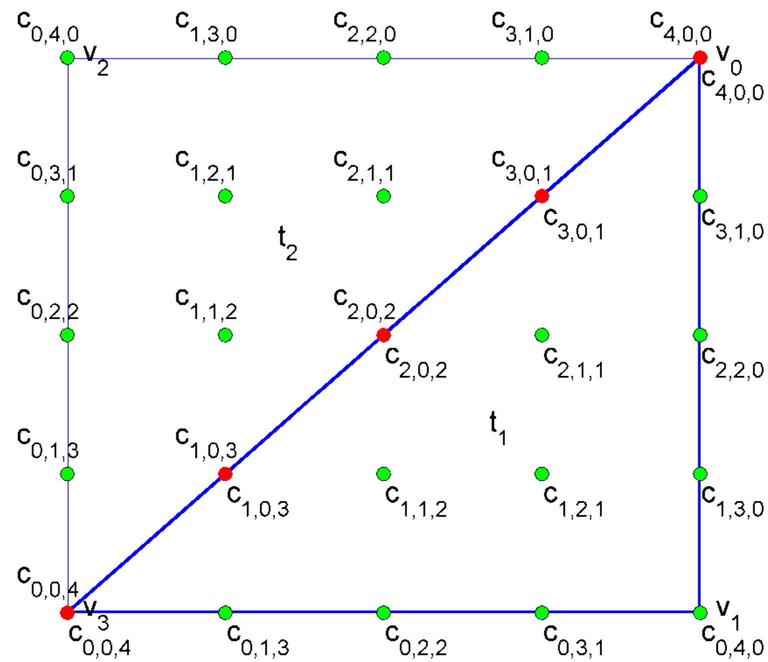
Continuity of Simplex B-Splines

Continuity between simplices: **Structure of Continuity**

$$c^{t_2}_{\kappa_0,m,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_*), \quad 0 \le m \le r$$



No continuity

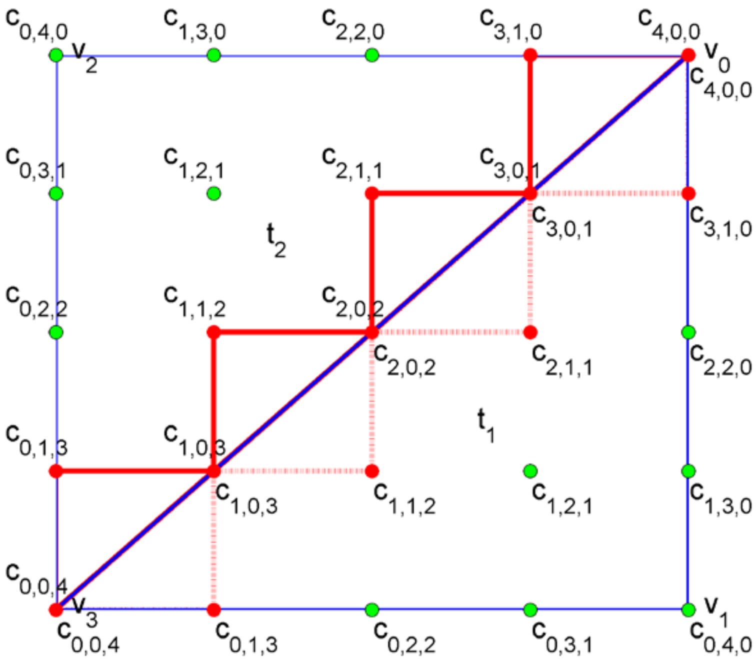


0th order continuity

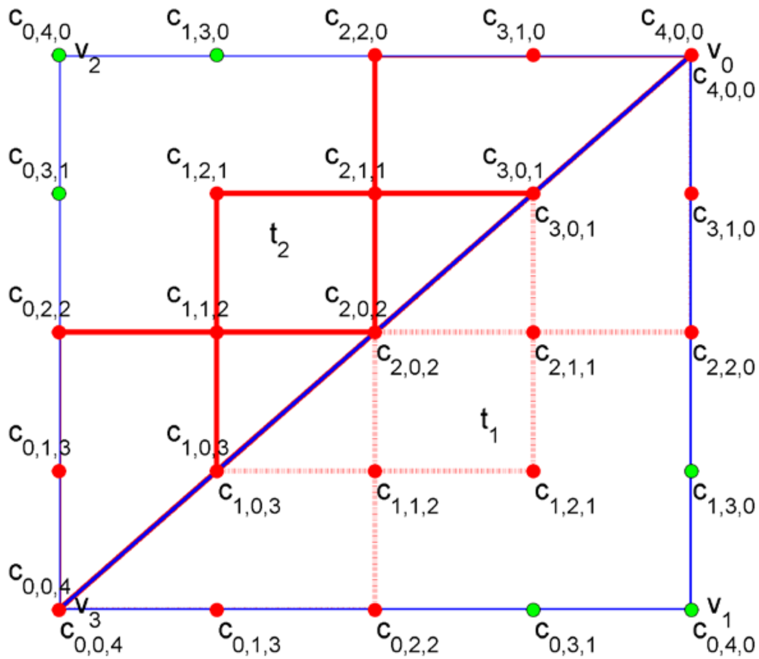
Continuity of Simplex B-Splines

Continuity between simplices: **Structure of Continuity**

$$c^{t_2}_{\kappa_0,m,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_*), \quad 0 \le m \le r$$



1st order continuity



2nd order continuity

Continuity of Simplex Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

We have:

$$c_{4,0,0}^{t_2} = c_{4,0,0}^{t_1}$$
$$c_{3,0,1}^{t_2} = c_{3,0,1}^{t_1}$$
$$c_{2,0,2}^{t_2} = c_{2,0,2}^{t_1}$$
$$c_{1,0,3}^{t_2} = c_{1,0,3}^{t_1}$$
$$c_{0,0,4}^{t_2} = c_{0,0,4}^{t_1}$$

0th order continuity

$$c_{3,1,0}^{t_2} = c_{4,0,0}^{t_1} \cdot b_0(v_2) + c_{3,1,0}^{t_1} \cdot b_1(v_2) + c_{3,0,1}^{t_1} \cdot b_2(v_2)$$
$$c_{2,1,1}^{t_2} = c_{3,0,1}^{t_1} \cdot b_0(v_2) + c_{2,1,1}^{t_1} \cdot b_1(v_2) + c_{2,0,2}^{t_1} \cdot b_2(v_2)$$
$$c_{1,1,2}^{t_2} = c_{2,0,2}^{t_1} \cdot b_0(v_2) + c_{1,1,2}^{t_1} \cdot b_1(v_2) + c_{1,0,3}^{t_1} \cdot b_2(v_2)$$
$$c_{0,1,3}^{t_2} = c_{1,0,3}^{t_1} \cdot b_0(v_2) + c_{0,1,3}^{t_1} \cdot b_1(v_2) + c_{0,0,4}^{t_1} \cdot b_2(v_2)$$

1st order continuity

Continuity of Simplex Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

Step 1: construct the **Global B-coefficient vector:**

$$c = \begin{bmatrix} c_{\kappa}^{t_1} \\ c_{\kappa}^{t_2} \end{bmatrix}_{|\kappa|=d}$$

Example: $d = 2$

$$c = \begin{bmatrix} c_{2,0,0}^{t_1} \\ c_{1,1,0}^{t_1} \\ c_{1,0,1}^{t_1} \\ c_{0,2,0}^{t_1} \\ c_{0,1,1}^{t_1} \\ c_{0,0,2}^{t_1} \\ c_{2,0,0}^{t_2} \\ c_{1,1,0}^{t_2} \\ c_{1,0,1}^{t_2} \\ c_{0,2,0}^{t_2} \\ c_{0,1,1}^{t_2} \\ c_{0,0,2}^{t_2} \end{bmatrix}$$

Continuity of Simplex Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

Step 2: write continuity equations in vector form:

$$c^{t_2}_{3,1,0} = c^{t_1}_{4,0,0} \cdot b_0(v_2) + c^{t_1}_{3,1,0} \cdot b_1(v_2) + c^{t_1}_{3,0,1} \cdot b_2(v_2)$$

subtract $c^{t_2}_{3,1,0}$

$$\longrightarrow 0 = c^{t_1}_{4,0,0} \cdot b_0(v_2) + c^{t_1}_{3,1,0} \cdot b_1(v_2) + c^{t_1}_{3,0,1} \cdot b_2(v_2) - c^{t_2}_{3,1,0}$$

$$\longrightarrow 0 = \Big[\underbrace{b_0(v_2) \quad b_1(v_2) \quad b_2(v_2) \quad \cdots \quad 0}_{\text{15 elements}} \quad \underbrace{-1 \quad \cdots \quad 0}_{\text{15 elements}} \Big] \cdot \begin{bmatrix} c^{t_1}_{\kappa} \\ c^{t_2}_{\kappa} \end{bmatrix}_{|\kappa|=d}$$

Continuity of Simplex Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

Step 3: compile the complete smoothness matrix:

$$\begin{bmatrix} b_0(v_2) & b_1(v_2) & b_2(v_2) & \cdots & 0 & -1 & \cdots & 0 \\ & & & \vdots & & & & \end{bmatrix} \cdot c = 0$$

All continuity equations taken are combined in the smoothness matrix, indicated as H :

$$\longrightarrow \quad H \cdot c = 0$$

With c the global vector of all B-coefficients:

Note: for r^{th} order continuity, we need to include **all** continuity equations $0,1,\ldots,r$

Continuity of Simplex Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

The smoothness matrix will get very large for large triangulations and continuity orders (see example).

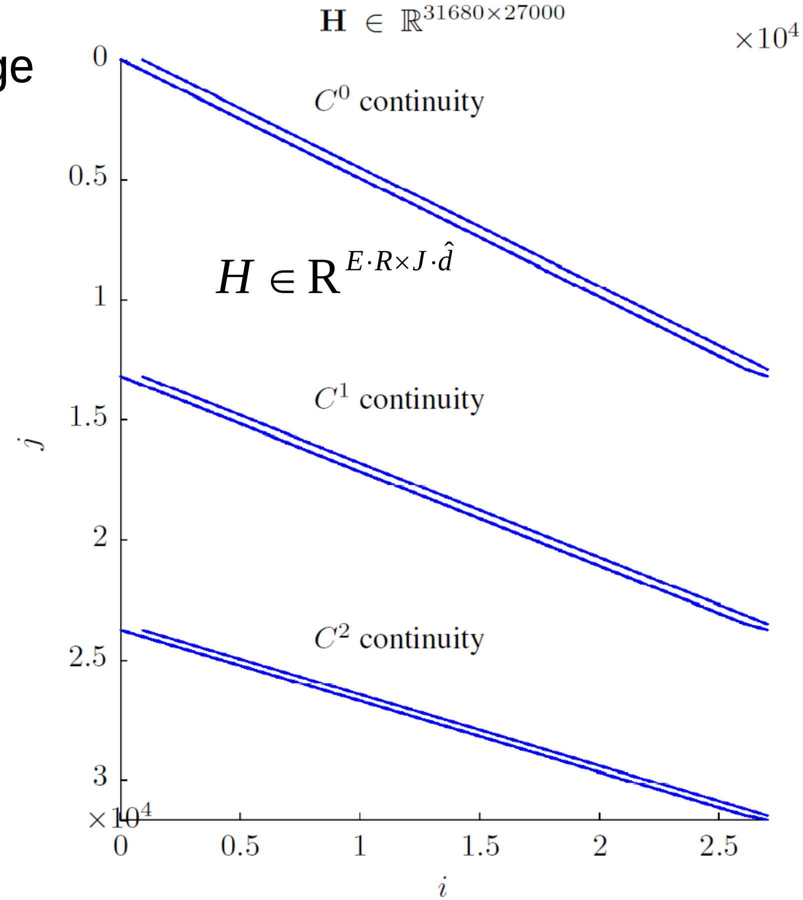
The total number of rows (=continuity equations) for H is:

$$row(H) = E \cdot R$$

with E the total number of edges in a triangulation, and R the total number of continuity conditions per edge.

In general, H will be rank deficient!

$$rank(H) < row(H)$$



H matrix for 1800 triangles, d=4, r=2

Continuity of Simplex B-Splines

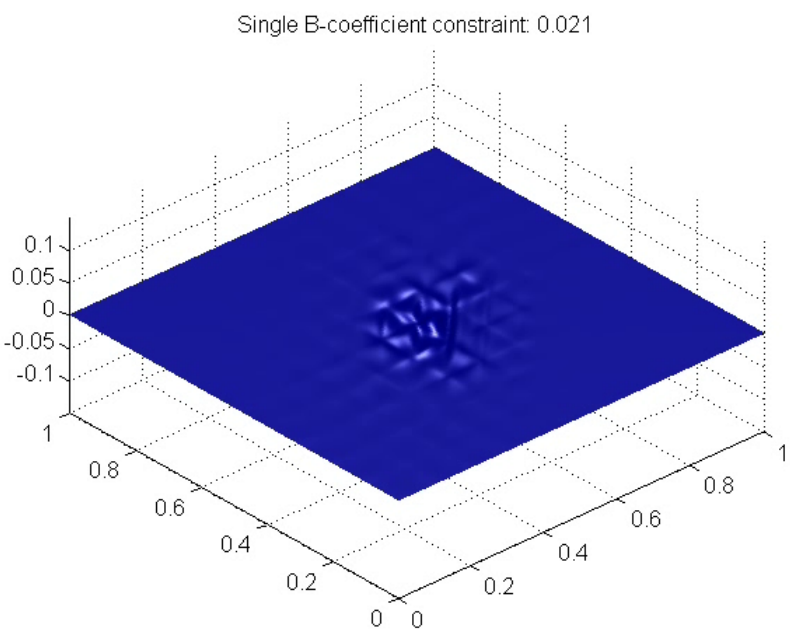
$for\ all\ edges\ in\ Triangulation\ do$ ...

$do\ for\ all\ continuity\ orders\ 0,1,\dots,m$

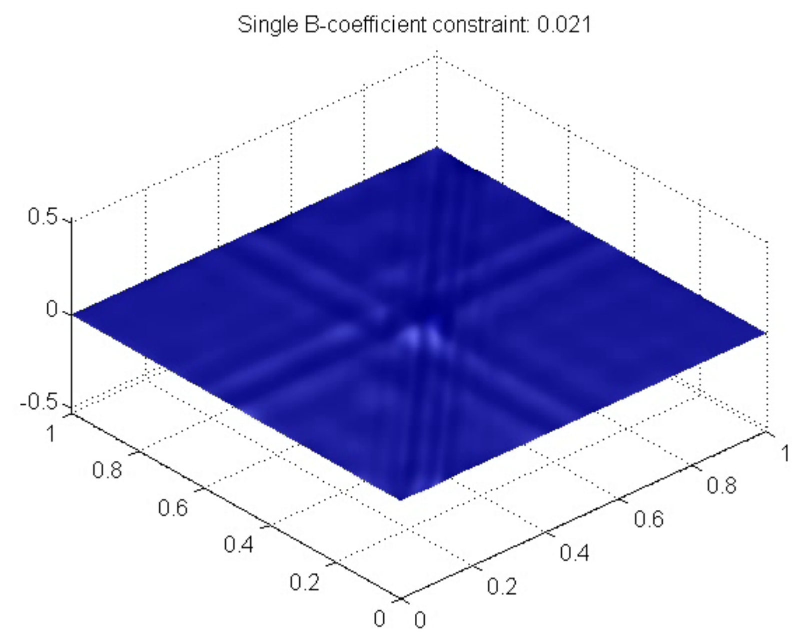
1. Left Hand part: Formulate valid permutations of multi-index
2. Right Hand part: Formulate valid permutations of γ with $|\gamma| = m$, expand $B_\gamma^m(v_*)$ if necessary
3. Right Hand part: Formulate valid permutations **using LH part permutations** together with permutations of γ.
4. Combine results: Formulate continuity conditions

Continuity of Simplex Splines

Continuity between simplices: Effects of continuity



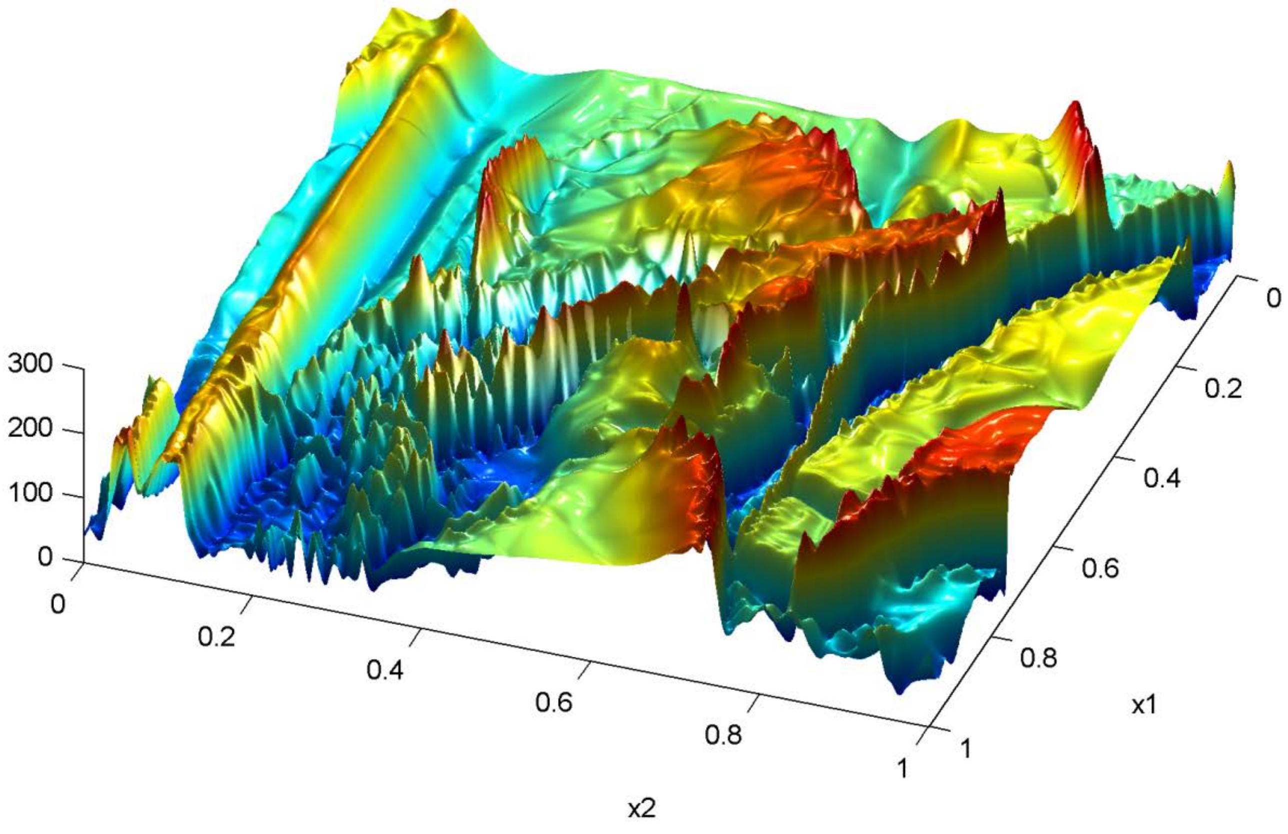
1st order continuity



2nd order continuity

In-depth: Continuity of Simplex B-Splines

Advanced discussion on Continuity

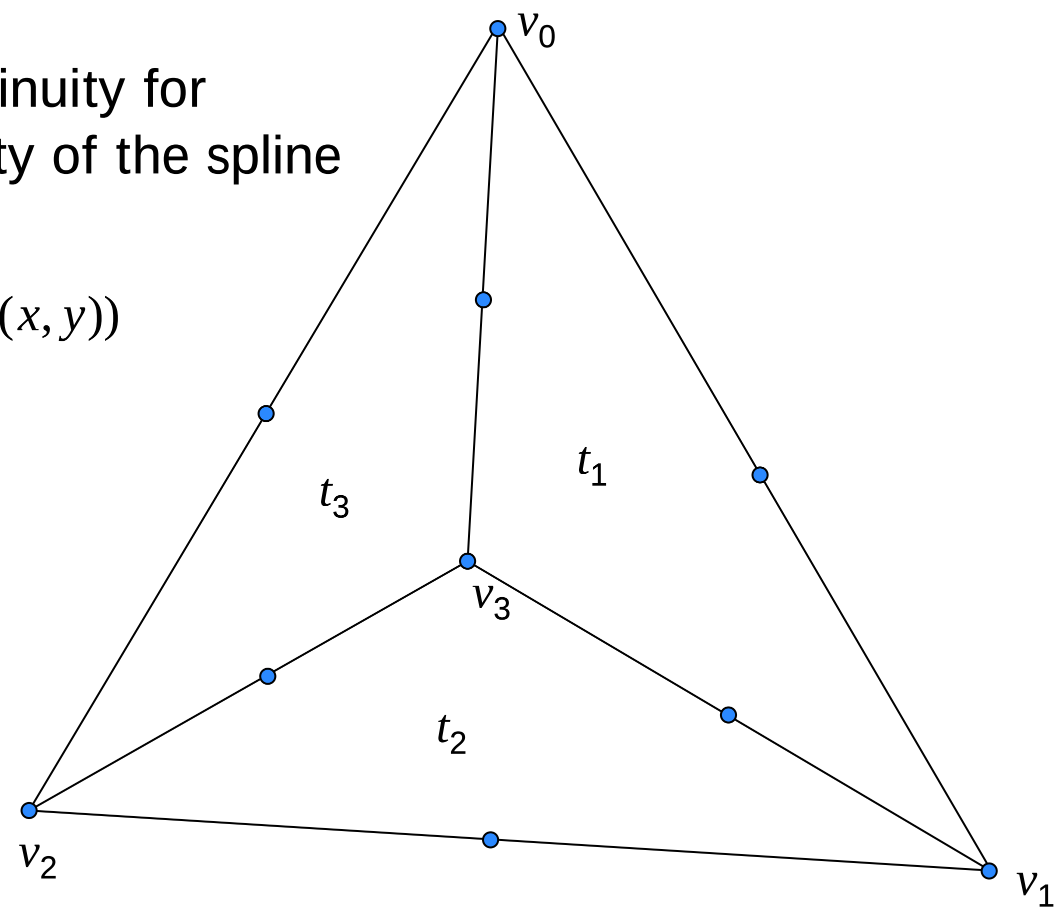


In-depth: Continuity of Simplex B-Splines

Exercise: structure of continuity

Plot the structure of continuity for 0th and 1st order continuity of the spline function:

$$p(b_t(x,y)) = \sum_{|\kappa|=2} c_\kappa^{t_j} B_\kappa^2(b_t(x,y))$$



B-net orientation rule:

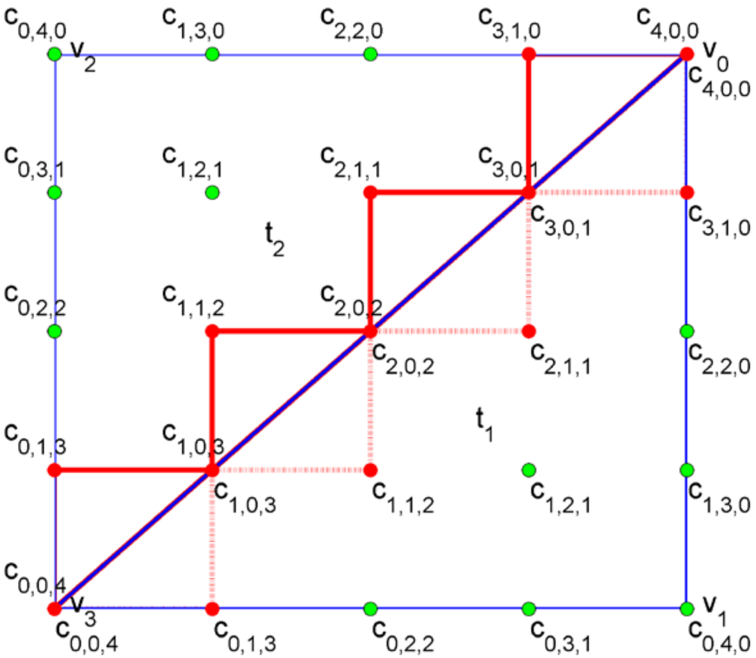
The B-coefficient with the **highest multi-index** value should be located at the vertex with the **lowest index**, the B-coefficient with the **second highest multi-index** value should be located at the vertex with the **second lowest index** and so on.

In-depth: Continuity of Simplex B-Splines

How to read the continuity conditions

The continuity conditions should be seen as a collection of equations relating B-coefficients...

$$\underbrace{c^{t_1}_{\kappa_0,m,\kappa_1}}_{\text{Left Hand part}} = \underbrace{\sum_{|\gamma|=m} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_1)}_{\text{Right Hand part}}$$



In-depth: Continuity of Simplex B-Splines

How to read the continuity conditions

$$c^{t_1}_{\kappa_{0_L},m,\kappa_{1_L}} = \sum_{|\gamma|=m} c^{t_2}_{(\kappa_{0_R},0,\kappa_{1_R})+\gamma} B^m_\gamma(v_1)$$

Left Hand part

Right Hand part

$$\kappa_{0_L} + m + \kappa_{1_L} = d \qquad (\kappa_{0_R} + 0 + \kappa_{1_R}) + \gamma = d$$

$$(\kappa_{0_R} + 0 + \kappa_{1_R}) + (\gamma_0 + \gamma_1 + \gamma_2) = d$$

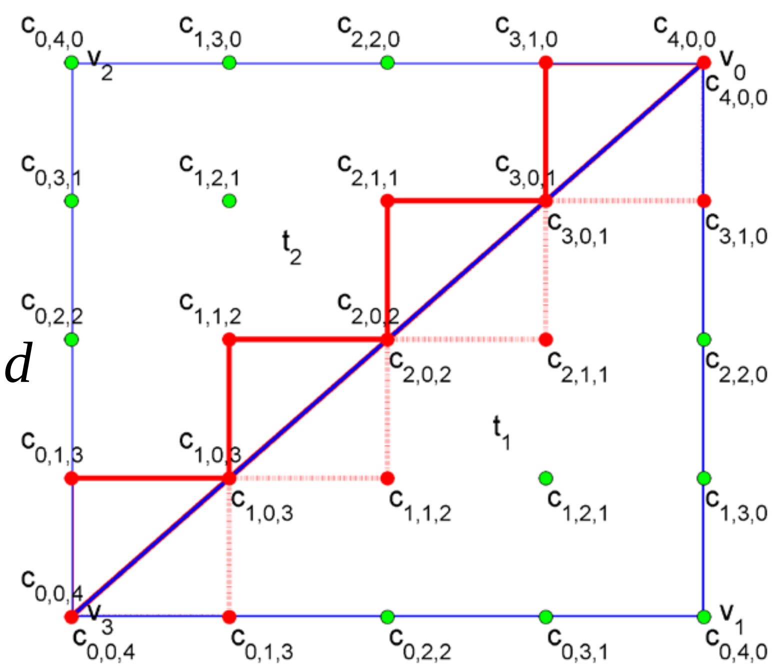
Very important:

$\kappa_{0_L} = \kappa_{0_R}$

$\kappa_{1_L} = \kappa_{1_R}$

M

$\kappa_{n_L} = \kappa_{n_R}$



In-depth: Continuity of Simplex B-Splines

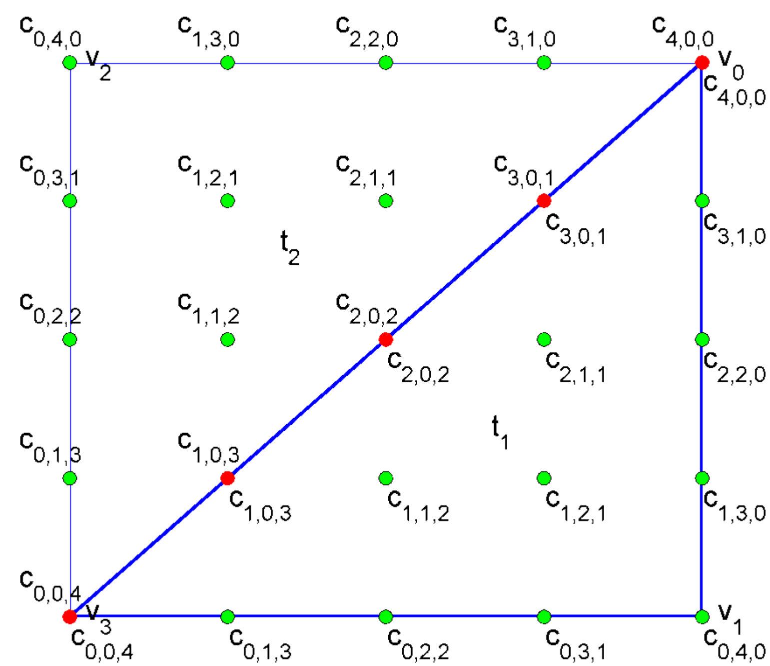
Example: How to read continuity conditions: d=4, m=0

$$\underbrace{c^{t_1}_{\kappa_0,0,\kappa_1}}_{\text{Left Hand part}} = \underbrace{\sum_{|\gamma|=0} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^0_\gamma(v_1)}_{\text{Right Hand part}}$$

$$\kappa_0 + 0 + \kappa_1 = 4$$

$$(\kappa_0, 0, \kappa_1) \in \begin{Bmatrix} (4,0,0) \\ (3,0,1) \\ (2,0,2) \\ (1,0,3) \\ (0,0,4) \end{Bmatrix}$$

Step 1: write out all valid permutations of LH part



In-depth: Continuity of Simplex B-Splines

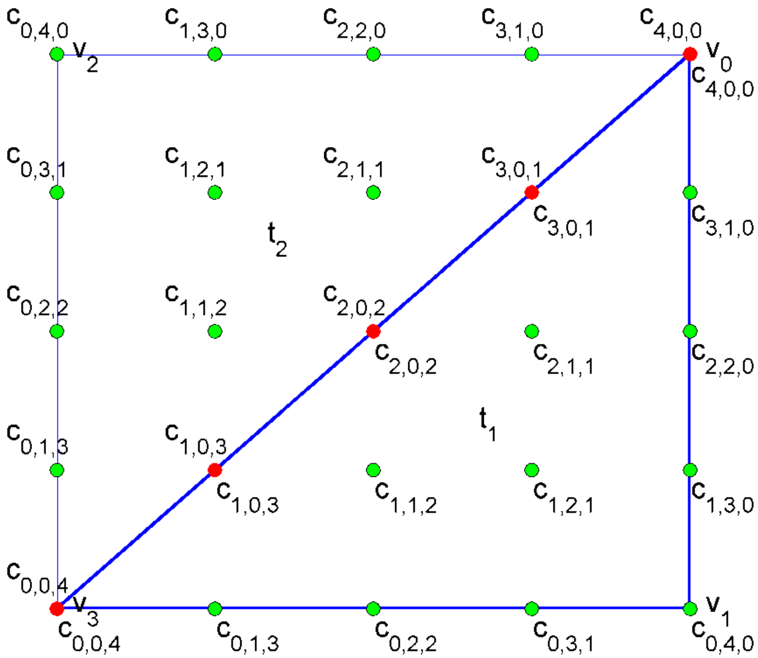
Example: How to read continuity conditions: d=4, m=0

$$\underbrace{c^{t_1}_{\kappa_0,0,\kappa_1}}_{\text{Left Hand part}} = \underbrace{\sum_{|\gamma|=0} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^0_\gamma(v_1)}_{\text{Right Hand part}}$$

$$\gamma \in \{(0,0,0)\}$$

$$B^0_\gamma(v_1) = 1$$

Step 2: write out all valid permutations of γ with $|\gamma| = m$, expand $B^0_\gamma(v_2)$ if necessary.



In-depth: Continuity of Simplex B-Splines

Example: How to read continuity conditions: d=4, m=0

$$\underbrace{c^{t_1}_{\kappa_0,0,\kappa_1}}_{\text{Left Hand part}} = \underbrace{\sum_{|\gamma|=0} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^0_\gamma(v_1)}_{\text{Right Hand part}}$$

Step 3: write out all valid permutations of RH part **using** LH part permutations and permutations of γ.

$$(\kappa_0,0,\kappa_1) \in \begin{Bmatrix} (4,0,0) \\ (3,0,1) \\ (2,0,2) \\ (1,0,3) \\ (0,0,4) \end{Bmatrix}$$

$$\gamma \in \{(0,0,0)\}$$

$$(\kappa_0,0,\kappa_1)+\gamma \in \begin{Bmatrix} (4,0,0)+(0,0,0) \\ (3,0,1)+(0,0,0) \\ (2,0,2)+(0,0,0) \\ (1,0,3)+(0,0,0) \\ (0,0,4)+(0,0,0) \end{Bmatrix} = \begin{Bmatrix} (4,0,0) \\ (3,0,1) \\ (2,0,2) \\ (1,0,3) \\ (0,0,4) \end{Bmatrix}$$

value taken from κ_0 in LH part

value taken from κ_1 in LH part

In-depth: Continuity of Simplex B-Splines

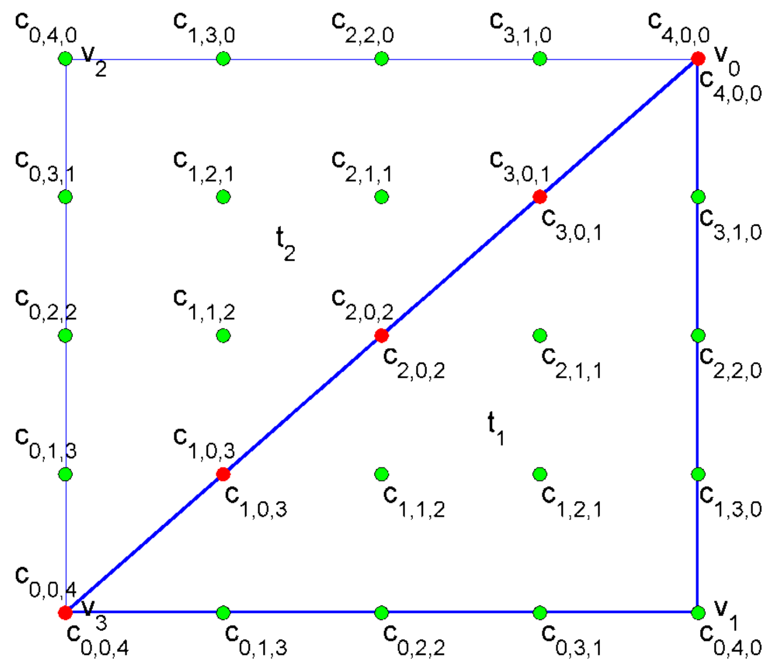
Example: How to read continuity conditions: d=4, m=0

$$(\kappa_0, 0, \kappa_1) \in \left\{ \begin{array}{c} (4,0,0) \\ (3,0,1) \\ (2,0,2) \\ (1,0,3) \\ (0,0,4) \end{array} \right\}$$

$$(\kappa_0, 0, \kappa_1) + \gamma \in \left\{ \begin{array}{c} (4,0,0) \\ (3,0,1) \\ (2,0,2) \\ (1,0,3) \\ (0,0,4) \end{array} \right\}$$

$$\underbrace{c^{t_1}_{\kappa_0,0,\kappa_1}} = \underbrace{\sum_{|\gamma|=0} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^0_\gamma(v_1)}$$

Step 4: construct smoothness conditions, and check structure of continuity.



Smoothness conditions for d=4, m=0

$$\begin{array}{|c|} \hline c^{t_1}_{4,0,0} = c^{t_2}_{4,0,0} \\ c^{t_1}_{3,0,1} = c^{t_2}_{3,0,1} \\ c^{t_1}_{2,0,2} = c^{t_2}_{2,0,2} \\ c^{t_1}_{1,0,3} = c^{t_2}_{1,0,3} \\ c^{t_1}_{0,0,4} = c^{t_2}_{0,0,4} \\ \hline \end{array}$$

In-depth: Continuity of Simplex B-Splines

$for\ all\ edges\ in\ Triangulation\ do\ ...$

$do\ for\ all\ continuity\ orders\ 0,1,\dots,m$

1. Left Hand part: Formulate valid permutations of multi-index
2. Right Hand part: Formulate valid permutations of γ with $|\gamma| = m$, expand $B_\gamma^m(v_*)$ if necessary
3. Right Hand part: Formulate valid permutations **using LH part permutations** together with permutations of γ.
4. Combine results: Formulate continuity conditions

In-depth: Continuity of Simplex B-Splines

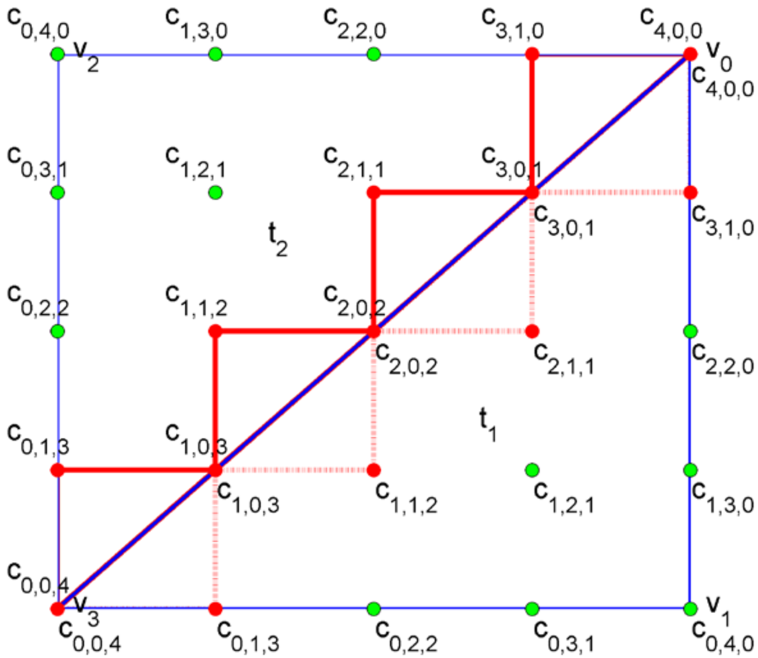
Example: Higher order continuity d=4, m=1

$$\underbrace{c^{t_1}_{\kappa_0,1,\kappa_1}}_{\text{Left Hand part}} = \underbrace{\sum_{|\gamma|=1} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^1_\gamma(v_1)}_{\text{Right Hand part}}$$

$$\kappa_0 + 1 + \kappa_1 = 4$$

$$(\kappa_0, 1, \kappa_1) \in \begin{Bmatrix} (3,1,0) \\ (2,1,1) \\ (1,1,2) \\ (0,1,3) \end{Bmatrix}$$

Step 1: write out all valid permutations of LH part



In-depth: Continuity of Simplex B-Splines

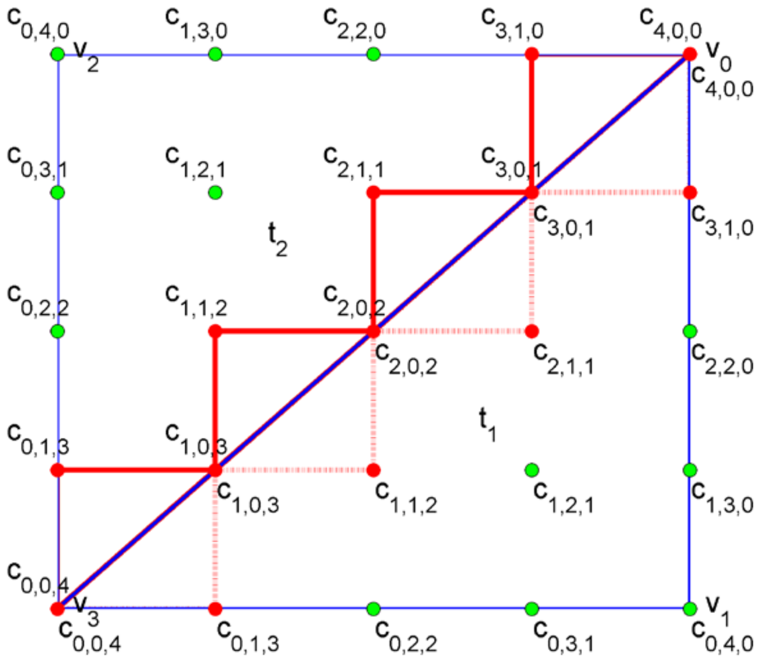
Example: Higher order continuity d=4, m=1

$$\underbrace{c^{t_1}_{\kappa_0,1,\kappa_1}}_{\text{Left Hand part}} = \underbrace{\sum_{|\gamma|=1} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^1_\gamma(v_1)}_{\text{Right Hand part}}$$

$$\gamma \in \begin{Bmatrix} (1,0,0) \\ (0,1,0) \\ (0,0,1) \end{Bmatrix}$$

$$B^1_\gamma(v_1) \in \{B^1_{100}(v_1), B^1_{010}(v_1), B^1_{001}(v_1)\}$$

Step 2: write out all valid permutations of γ with $|\gamma| = m$, expand $B^0_\gamma(v_2)$ if necessary.



In-depth: Continuity of Simplex B-Splines

Example: Higher order continuity d=4, m=1

$$\underbrace{c^{t_1}_{\kappa_0,1,\kappa_1}}_{\text{Left Hand part}} = \underbrace{\sum_{|\gamma|=1} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^1_\gamma(v_1)}_{\text{Right Hand part}}$$

Step 3: write out all valid permutations of RH part **using** LH part permutations and permutations of γ.

$$(\kappa_0, 1, \kappa_1) \in \begin{Bmatrix} (3,1,0) \\ (2,1,1) \\ (1,1,2) \\ (0,1,3) \end{Bmatrix}$$

$$\gamma \in \{(1,0,0),(0,1,0),(0,0,1)\}$$

$$(\kappa_0, 0, \kappa_1) + \gamma \in \begin{Bmatrix} (3,0,0)+\gamma \\ (2,0,1)+\gamma \\ (1,0,2)+\gamma \\ (0,0,3)+\gamma \end{Bmatrix}$$

value taken from κ_0 in LH part

value taken from κ_1 in LH part

In-depth: Continuity of Simplex B-Splines

Example: Higher order continuity d=4, m=1

Step 3 (continued)

$$\gamma \in \{(1,0,0),(0,1,0),(0,0,1)\}$$

$$(\kappa_0, 0, \kappa_1) + \gamma \in \begin{Bmatrix} (3,0,0)+\gamma \\ (2,0,1)+\gamma \\ (1,0,2)+\gamma \\ (0,0,3)+\gamma \end{Bmatrix} = \left\{ \begin{array}{l} (3,0,0)+\gamma = \begin{cases} (3,0,0)+(1,0,0)=(4,0,0) \\ (3,0,0)+(0,1,0)=(3,1,0) \\ (3,0,0)+(0,0,1)=(3,0,1) \end{cases} \\ (2,0,1)+\gamma = \begin{cases} (2,0,1)+(1,0,0)=(3,0,1) \\ (2,0,1)+(0,1,0)=(2,1,1) \\ (2,0,1)+(0,0,1)=(2,0,2) \end{cases} \\ (1,0,2)+\gamma = \begin{cases} (1,0,2)+(1,0,0)=(2,0,2) \\ (1,0,2)+(0,1,0)=(1,1,2) \\ (1,0,2)+(0,0,1)=(1,0,3) \end{cases} \\ (0,0,3)+\gamma = \begin{cases} (0,0,3)+(1,0,0)=(1,0,3) \\ (0,0,3)+(0,1,0)=(0,1,3) \\ (0,0,3)+(0,0,1)=(0,0,4) \end{cases} \end{array} \right.$$

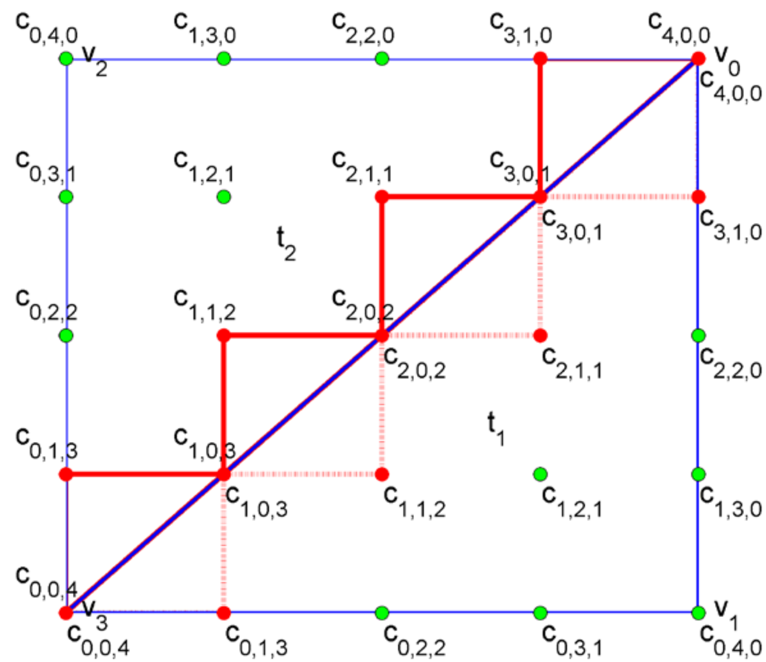
In-depth: Continuity of Simplex B-Splines

Example: Higher order continuity d=4, m=1

$$(\kappa_0, 1, \kappa_1) \in \begin{Bmatrix} (3,1,0) \\ (2,1,1) \\ (1,1,2) \\ (0,1,3) \end{Bmatrix} \qquad (\kappa_0, 0, \kappa_1) + \gamma \in \begin{Bmatrix} (3,0,0)+\gamma \\ (2,0,1)+\gamma \\ (1,0,2)+\gamma \\ (0,0,3)+\gamma \end{Bmatrix}$$

$$c^{t_1}_{\kappa_0,1,\kappa_1} = \sum_{|\gamma|=1} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^1_\gamma(v_1)$$

Step 4: construct smoothness conditions, and check structure of continuity.



Smoothness conditions for d=4, m=1

$$\begin{aligned}
c^{t_1}_{3,1,0} &= c^{t_2}_{4,0,0}B^1_{100}(v_1) + c^{t_2}_{3,1,0}B^1_{010}(v_1) + c^{t_2}_{3,0,1}B^1_{001}(v_1) \\
c^{t_1}_{2,1,1} &= c^{t_2}_{3,0,1}B^1_{100}(v_1) + c^{t_2}_{2,1,1}B^1_{010}(v_1) + c^{t_2}_{2,0,2}B^1_{001}(v_1) \\
c^{t_1}_{1,1,2} &= c^{t_2}_{2,0,2}B^1_{100}(v_1) + c^{t_2}_{1,1,2}B^1_{010}(v_1) + c^{t_2}_{1,0,3}B^1_{001}(v_1) \\
c^{t_1}_{0,1,3} &= c^{t_2}_{1,0,3}B^1_{100}(v_1) + c^{t_2}_{0,1,3}B^1_{010}(v_1) + c^{t_2}_{0,0,4}B^1_{001}(v_1)
\end{aligned}$$

$$(3,0,0)+\gamma = \begin{cases} (4,0,0) \\ (3,1,0) \\ (3,0,1) \end{cases} \quad (2,0,1)+\gamma = \begin{cases} (3,0,1) \\ (2,1,1) \\ (2,0,2) \end{cases} \quad (1,0,2)+\gamma = \begin{cases} (2,0,2) \\ (1,1,2) \\ (1,0,3) \end{cases} \quad (0,0,3)+\gamma = \begin{cases} (1,0,3) \\ (0,1,3) \\ (0,0,4) \end{cases}$$

In-depth: Continuity of Simplex B-Splines

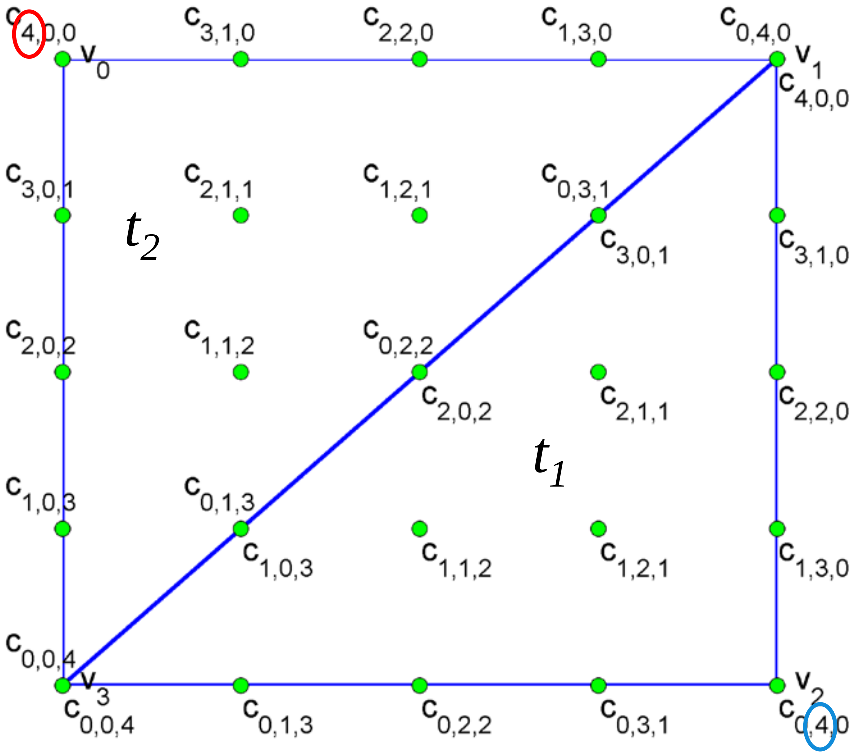
Generalizing Continuity between simplices

The location of the ‘m’ and ‘0’ in the multi-index of the continuity equations is equal to the location of the single non-zero in the multi-index at the out-of-edge vertices!

$$c^{t_1}_{\kappa_0,m,\kappa_1} = \sum_{|\gamma|=m} c^{t_2}_{(0,\kappa_0,\kappa_1)+\gamma} B^m_\gamma(v_2)$$

or alternatively, writing continuity from t_2 to t_1:

$$c^{t_2}_{m,\kappa_0,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_0)$$



Generalizing the equations of continuity

The location of the constant ‘m’ and ‘0’ within the multi-index of the B-coefficients is determined by the location of the non-zero value in the multi-index of B-coefficients at the out-of-edge vertices of the respective simplices.

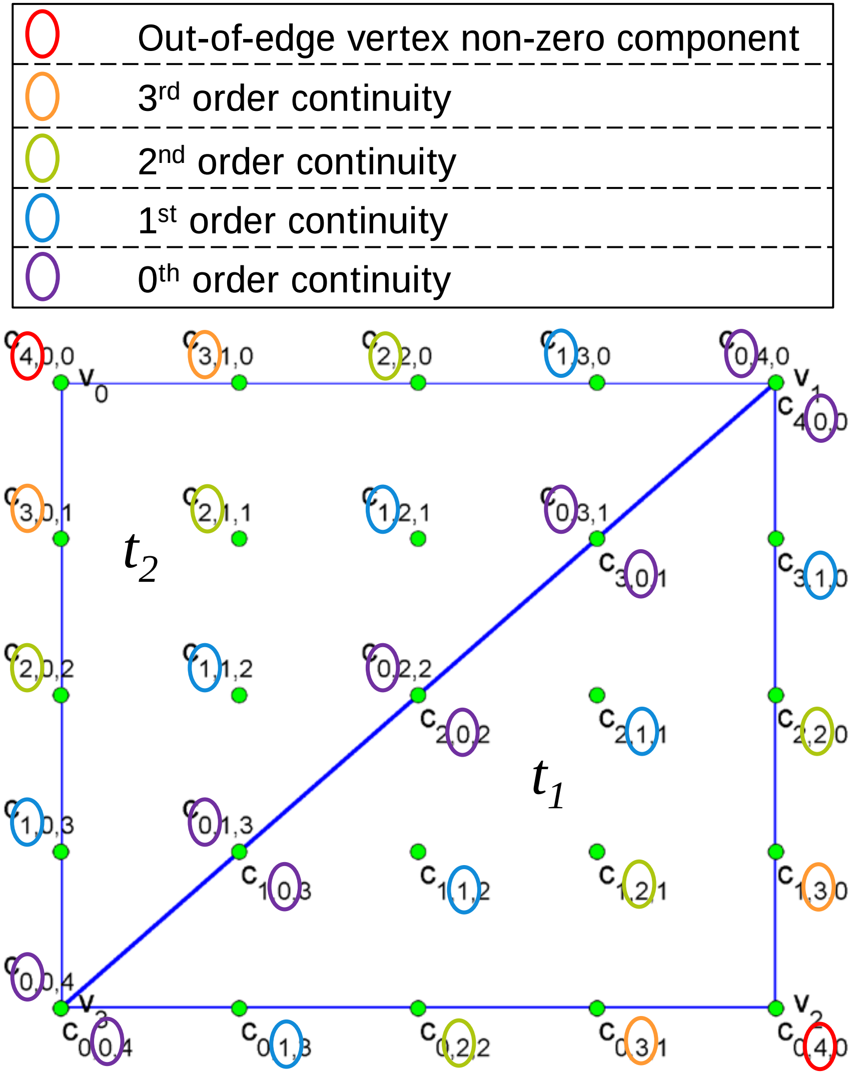
In-depth: Continuity of Simplex B-Splines

Generalizing Continuity

Notice that the single non-zero multi-index component at the out-of-edge vertices has a value ‘d’ ($\kappa_0 = 4$ for t_2, $\kappa_1 = 4$ for t_1 in example right).

When ‘marching’ from the out-of-edge-vertex to the edge, the value of this component reduces from ‘d’ to 0.

The value of this component also indicates the order of continuity of which it will be a part. (see right)



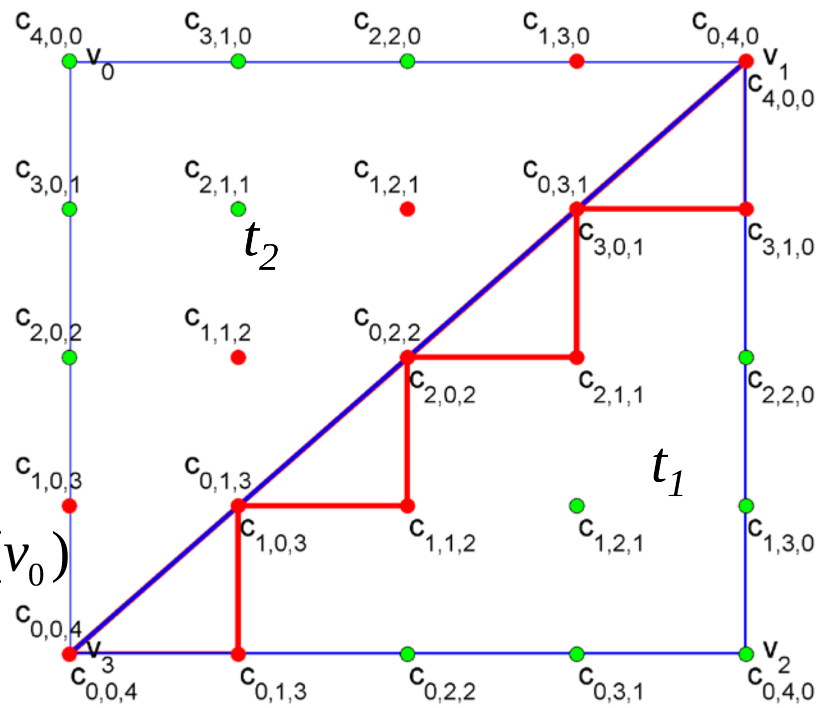
In-depth: Continuity of Simplex B-Splines

Example 6.7: First order continuity between two simplices revisited

$$c^{t_2}_{m,\kappa_0,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_0), \quad 0 \le m \le 1$$

For example, for $c^{t_2}_{1,3,0}$ we get:

$$\begin{aligned} c^{t_2}_{1,3,0} &= c^{t_1}_{(3,0,0)+(1,0,0)} B^1_{1,0,0}(v_0) + \\ &\quad c^{t_1}_{(3,0,0)+(0,1,0)} B^1_{0,1,0}(v_0) + \\ &\quad c^{t_1}_{(3,0,0)+(0,0,1)} B^1_{0,0,1}(v_0) \\ &= c^{t_1}_{4,0,0} B^1_{1,0,0}(v_0) + c^{t_1}_{3,1,0} B^1_{0,1,0}(v_0) + c^{t_1}_{3,0,1} B^1_{0,0,1}(v_0) \\ &= c^{t_1}_{4,0,0} \cdot b_0(v_0) + c^{t_1}_{3,1,0} \cdot b_1(v_0) + c^{t_1}_{3,0,1} \cdot b_2(v_0) \end{aligned}$$



1st order continuity $(r = 1)$

In-depth: Continuity of Simplex B-Splines

Example 6.7: First order continuity between two simplices revisited

$$c^{t_2}_{m,\kappa_0,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_0), \quad 0 \le m \le 1$$

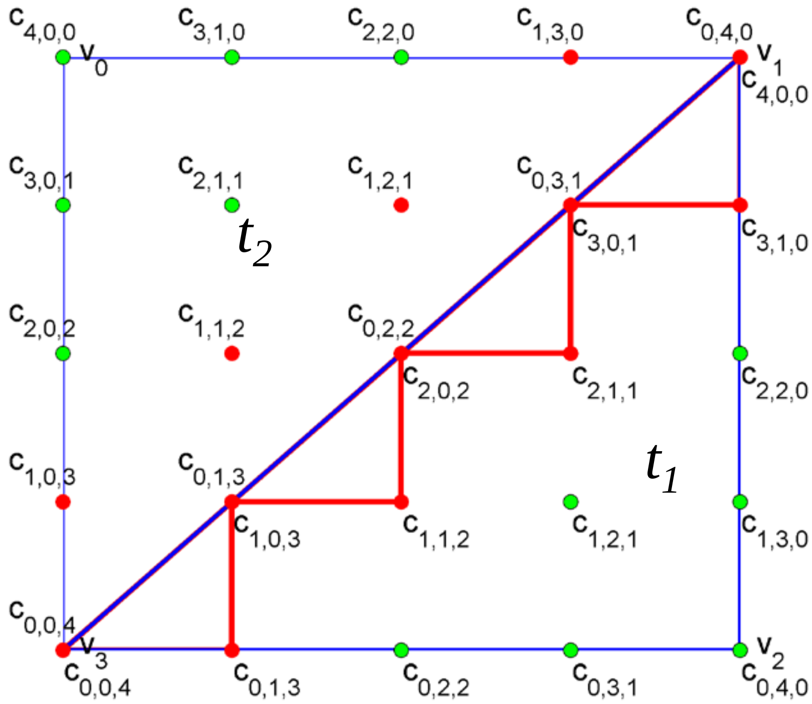
1st order continuity equations:

$$c^{t_2}_{1,3,0} = c^{t_1}_{4,0,0} \cdot b_0(v_0) + c^{t_1}_{3,1,0} \cdot b_1(v_0) + c^{t_1}_{3,0,1} \cdot b_2(v_0)$$

$$c^{t_2}_{1,2,1} = c^{t_1}_{3,0,1} \cdot b_0(v_0) + c^{t_1}_{2,1,1} \cdot b_1(v_0) + c^{t_1}_{2,0,2} \cdot b_2(v_0)$$

$$c^{t_2}_{1,1,2} = c^{t_1}_{2,0,2} \cdot b_0(v_0) + c^{t_1}_{1,1,2} \cdot b_1(v_0) + c^{t_1}_{1,0,3} \cdot b_2(v_0)$$

$$c^{t_2}_{1,0,3} = c^{t_1}_{1,0,3} \cdot b_0(v_0) + c^{t_1}_{0,1,3} \cdot b_1(v_0) + c^{t_1}_{0,0,4} \cdot b_2(v_0)$$



1st order continuity $(r = 1)$

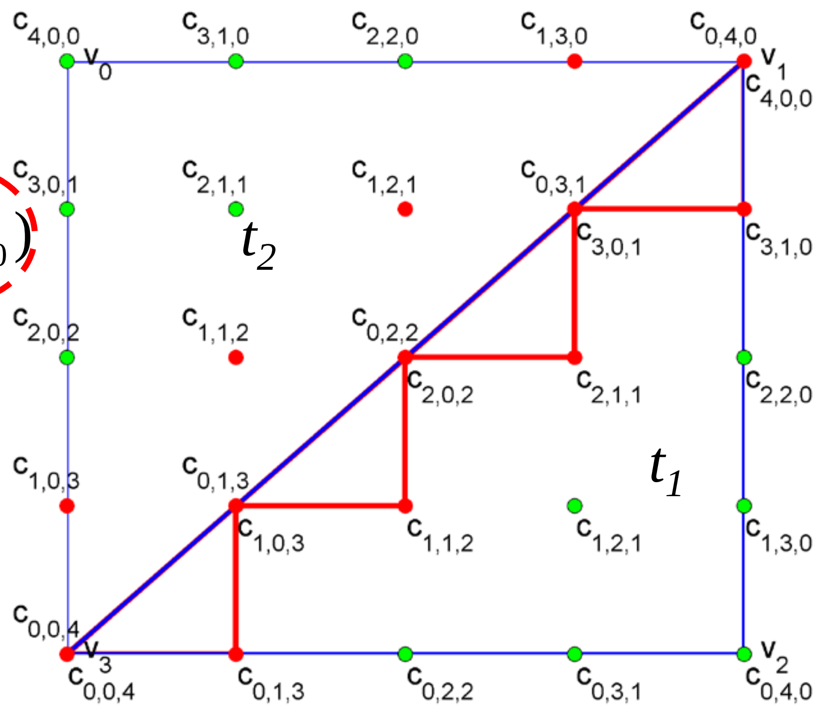
In-depth: Continuity of Simplex B-Splines

Example 6.7: First order continuity between two simplices revisited

$$c^{t_2}_{m,\kappa_0,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_0), \quad 0 \le m \le 1$$

1st order continuity equation:

$$c^{t_2}_{1,3,0} = c^{t_1}_{4,0,0} \cdot b_0(v_0) + c^{t_1}_{3,1,0} \cdot b_1(v_0) + c^{t_1}_{3,0,1} \cdot b_2(v_0)$$



1st order continuity ($r = 1$)

Recall:

inside simplex $b_0 \ge 0, b_1 \ge 0, b_2 \ge 0$

but barycentric coordinates of v_0 are taken with respect to t_1...

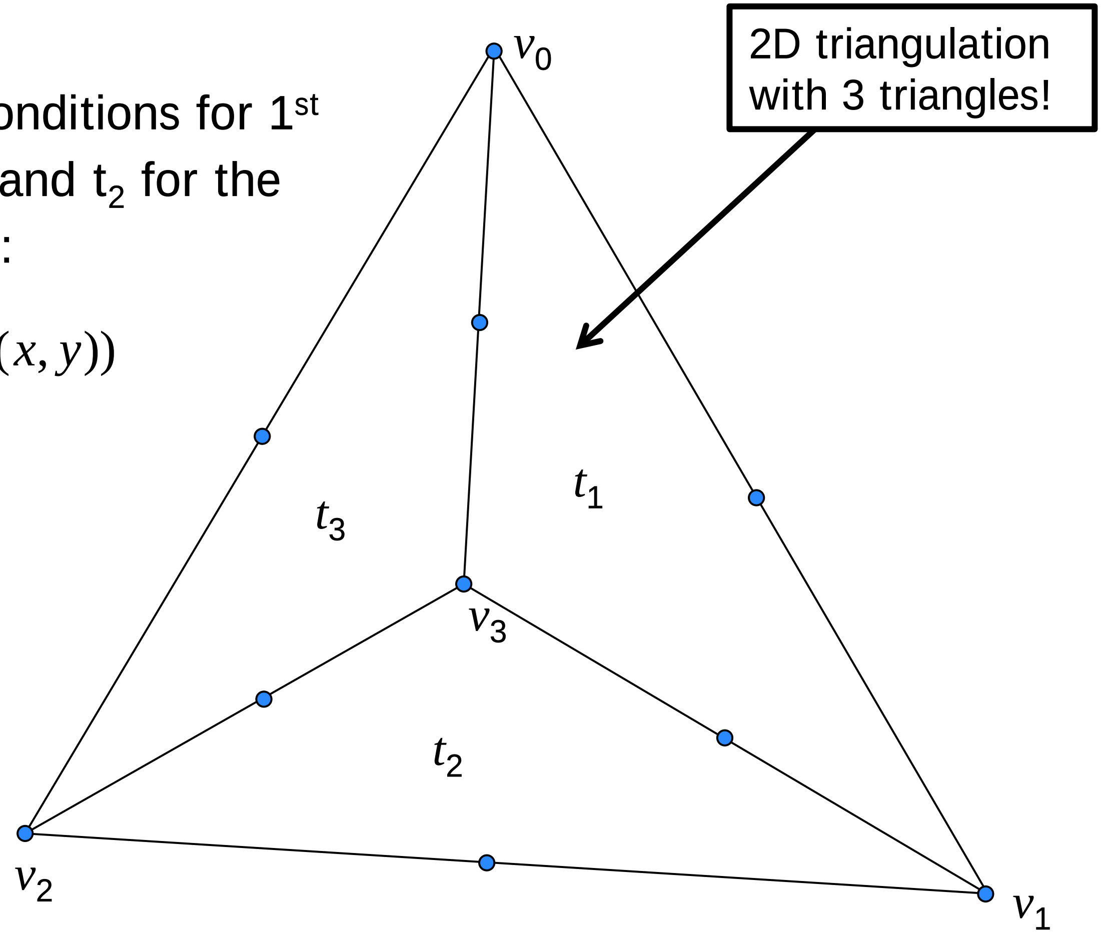
So in general at least one $b_0(v_0), b_1(v_0), b_2(v_0) < 0$!

In-depth: Continuity of Simplex B-Splines

Exercise: continuity conditions

Write down the continuity conditions for 1st order continuity between t_1 and t_2 for the quadratic 2D spline function:

$$p(b_t(x,y)) = \sum_{|\kappa|=2} c_\kappa^{t_j} B_\kappa^2(b_t(x,y))$$



Use either:

$$c_{m,\kappa_0,\kappa_1}^{t_1} = \sum_{|\gamma|=m} c_{(\kappa_0,0,\kappa_1)+\gamma}^{t_2} B_\gamma^m(v_0)$$

$$c_{\kappa_0,m,\kappa_1}^{t_2} = \sum_{|\gamma|=m} c_{(0,\kappa_0,\kappa_1)+\gamma}^{t_1} B_\gamma^m(v_2)$$

In-depth: Continuity of Simplex B-Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

We have:

$$\left.\begin{aligned}
c^{t_2}_{0,4,0} &= c^{t_1}_{4,0,0}\\
c^{t_2}_{0,3,1} &= c^{t_1}_{3,0,1}\\
c^{t_2}_{0,2,2} &= c^{t_1}_{2,0,2}\\
c^{t_2}_{0,1,3} &= c^{t_1}_{1,0,3}\\
c^{t_2}_{0,0,4} &= c^{t_1}_{0,0,4}
\end{aligned}\right\} \rightarrow \text{0}^{\text{th}}\text{ order continuity}$$

$$\left.\begin{aligned}
c^{t_2}_{1,3,0} &= c^{t_1}_{4,0,0}\cdot b_0(v_0) + c^{t_1}_{3,1,0}\cdot b_1(v_0) + c^{t_1}_{3,0,1}\cdot b_2(v_0)\\
c^{t_2}_{1,2,1} &= c^{t_1}_{3,0,1}\cdot b_0(v_0) + c^{t_1}_{2,1,1}\cdot b_1(v_0) + c^{t_1}_{2,0,2}\cdot b_2(v_0)\\
c^{t_2}_{1,1,2} &= c^{t_1}_{2,0,2}\cdot b_0(v_0) + c^{t_1}_{1,1,2}\cdot b_1(v_0) + c^{t_1}_{1,0,3}\cdot b_2(v_0)\\
c^{t_2}_{1,0,3} &= c^{t_1}_{1,0,3}\cdot b_0(v_0) + c^{t_1}_{0,1,3}\cdot b_1(v_0) + c^{t_1}_{0,0,4}\cdot b_2(v_0)
\end{aligned}\right\} \rightarrow \text{1}^{\text{st}}\text{ order continuity}$$

In-depth: Continuity of Simplex B-Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

Step 1: construct the **Global B-coefficient vector:**

$$c = \begin{bmatrix} c_\kappa^{t_1} \\ c_\kappa^{t_2} \end{bmatrix}_{|\kappa|=d}$$

Example: $d = 2$

$$\longrightarrow \quad c = \begin{bmatrix} c_{2,0,0}^{t_1} \\ c_{1,1,0}^{t_1} \\ c_{1,0,1}^{t_1} \\ c_{0,2,0}^{t_1} \\ c_{0,1,1}^{t_1} \\ c_{0,0,2}^{t_1} \\ c_{2,0,0}^{t_2} \\ c_{1,1,0}^{t_2} \\ c_{1,0,1}^{t_2} \\ c_{0,2,0}^{t_2} \\ c_{0,1,1}^{t_2} \\ c_{0,0,2}^{t_2} \end{bmatrix}$$

In-depth: Continuity of Simplex B-Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

Step 2: write continuity equations in vector form:

$$c^{t_1}_{1,1,0} = c^{t_2}_{2,0,0} \cdot b_0(v_1) + c^{t_2}_{1,1,0} \cdot b_1(v_1) + c^{t_2}_{1,0,1} \cdot b_2(v_1)$$

subtract $c^{t_1}_{1,1,0}$

$$\longrightarrow \quad 0 = c^{t_2}_{2,0,0} \cdot b_0(v_1) + c^{t_2}_{1,1,0} \cdot b_1(v_1) + c^{t_2}_{1,0,1} \cdot b_2(v_1) - c^{t_1}_{1,1,0}$$

$$\longrightarrow \quad 0 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & b_0(v_1) & b_1(v_1) & b_2(v_1) & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c^{t_1}_{2,0,0} \\ c^{t_1}_{1,1,0} \\ c^{t_1}_{1,0,1} \\ c^{t_1}_{0,2,0} \\ c^{t_1}_{0,1,1} \\ c^{t_1}_{0,0,2} \\ c^{t_2}_{2,0,0} \\ c^{t_2}_{1,1,0} \\ c^{t_2}_{1,0,1} \\ c^{t_2}_{0,2,0} \\ c^{t_2}_{0,1,1} \\ c^{t_2}_{0,0,2} \end{bmatrix}$$

In-depth: Continuity of Simplex B-Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

Step 3: compile the complete smoothness matrix:

$$0 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & b_0(v_1) & b_1(v_1) & b_2(v_1) & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \end{bmatrix} \cdot c$$

All continuity equations taken are combined in the smoothness matrix, indicated as H :

$$\longrightarrow \quad 0 = H \cdot c$$

With c the global vector of all B-coefficients:

In-depth: Continuity of Simplex B-Splines

Continuity between simplices: Construction of the **Smoothness Matrix**

The smoothness matrix will get very large for large triangulations and continuity orders (see example).

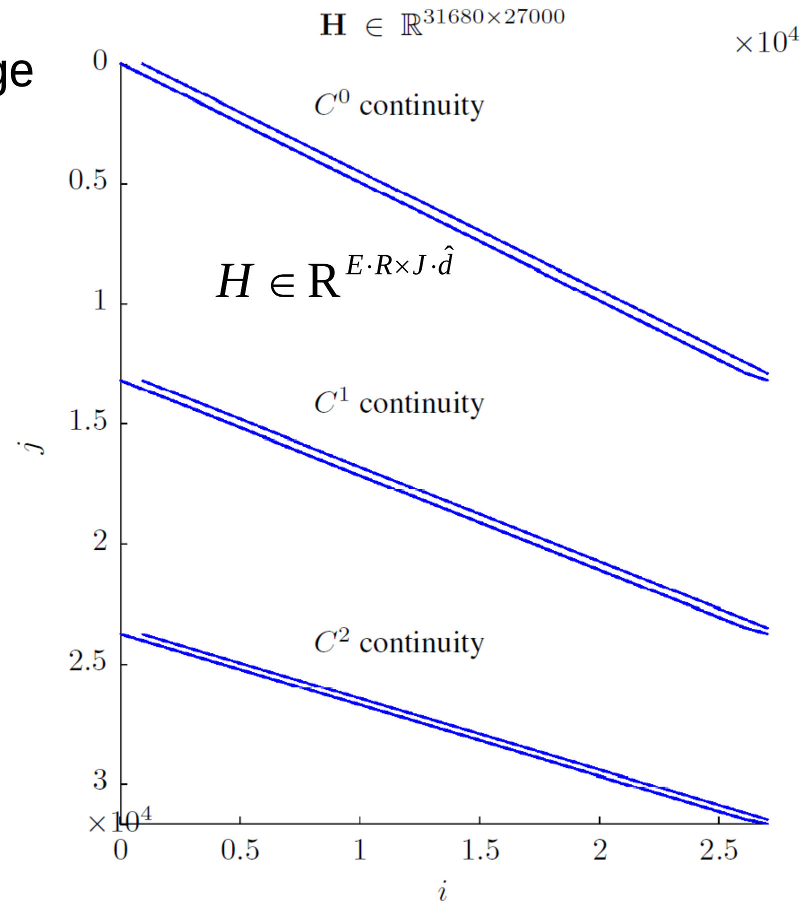
The total number of rows (=continuity equations) for H is:

$$row(H) = E \cdot R$$

with E the total number of edges in a triangulation, and R the total number of continuity conditions per edge.

In general, H will be rank deficient!

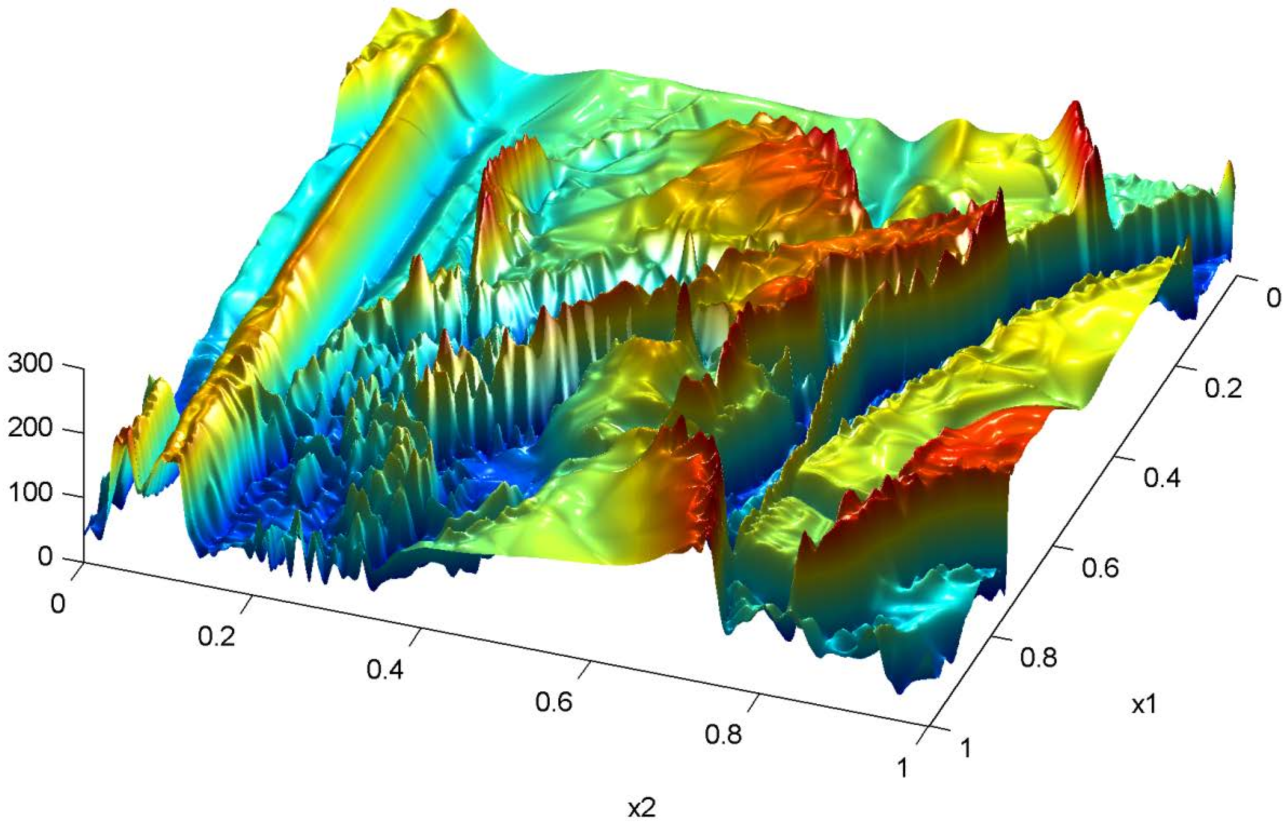
$$rank(H) < row(H)$$



H matrix for 1800 triangles, d=4, r=2

In-depth: Continuity of Simplex B-Splines

Finished: advanced discussion on Continuity



Simplex Spline Estimation

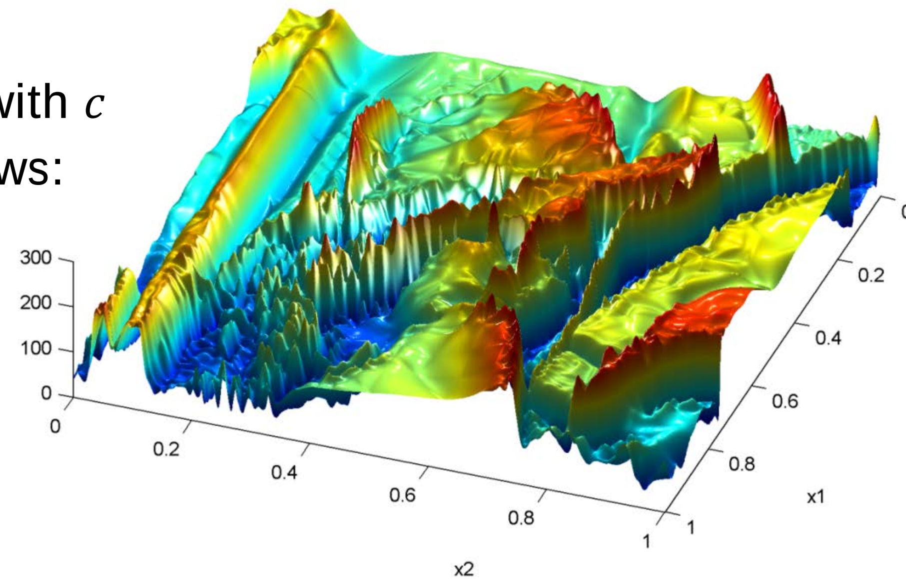
Formal definition of the multivariate simplex spline

We indicate a spline function of degree d and continuity order r on a triangulation T consisting of J simplices as follows:

$$s_r^d(x) = B \cdot c \quad \in \quad S_d^r(T_J)$$

With B the global regression matrix, with c the global B-coefficient vector as follows:

$$c = \begin{bmatrix} c_{t_1} \\ c_{t_2} \\ \vdots \\ c_{t_J} \end{bmatrix}$$



And with $S_d^r(T_J)$ the so-called "spline-space" of degree d and continuity order r on a triangulation T consisting of J simplices.

Simplex B-Spline Estimation

Formulation of the global B-form regression matrix

The global regression matrix B is constructed from per-simplex blocks B_{t_j} :

$$B = \begin{bmatrix} B_{t_1} & 0 & 0 & 0 \\ 0 & B_{t_2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & B_{t_J} \end{bmatrix} \in \mathbb{R}^{N \times J \cdot \hat{d}}$$

N_j: the per-simplex data count

N: total data count

With each per-simplex regression matrix block defined as follows B_{t_j} :

$$B_{t_j} = \begin{bmatrix} B^d_{d,0,0}(b_{t_j}(x(1))) & B^d_{d-1,1,0}(b_{t_j}(x(1))) & \cdots & B^d_{0,1,d-1}(b_{t_j}(x(1))) & B^d_{0,0,d}(b_{t_j}(x(1))) \\ B^d_{d,0,0}(b_{t_j}(x(2))) & B^d_{d-1,1,0}(b_{t_j}(x(2))) & \cdots & B^d_{0,1,d-1}(b_{t_j}(x(2))) & B^d_{0,0,d}(b_{t_j}(x(2))) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B^d_{d,0,0}(b_{t_j}(x(M))) & B^d_{d-1,1,0}(b_{t_j}(x(M))) & \cdots & B^d_{0,1,d-1}(b_{t_j}(x(M))) & B^d_{0,0,d}(b_{t_j}(x(M))) \end{bmatrix} \in \mathbb{R}^{N_j \times \hat{d}}$$

Simplex B-Spline Estimation

Equality constrained OLS estimation

We can now formulate the OLS cost function using the global regression matrix B and the global vector of B-coefficients c:

$$J(c) = \frac{1}{2}(Y - B \cdot c)^T (Y - B \cdot c)$$

We want to enforce smoothness between spline pieces using the smoothness constraints $H \cdot c = 0$.

The **constrained** cost function minimum can be found as follows:

$$\hat{c} = \arg\min \left[\frac{1}{2}(Y - B \cdot c)^T (Y - B \cdot c) \right], \quad \text{subject to } \; H \cdot c = 0$$

This equality constrained optimization problem can be solved using many different methods.

We will use a **Lagrange Multiplier method**…

Simplex B-Spline Estimation

Equality constrained OLS estimation

We want to solve the constrained OLS optimization problem:

$$\hat{c} = \arg\min\left[\frac{1}{2}\left(Y - B\cdot c\right)^T\left(Y - B\cdot c\right)\right], \quad \text{subject to } \; H\cdot c = 0$$

For this we formulate the **Lagrangian** $L(c,\lambda)$ as follows:

$$L(c,\lambda) = \frac{1}{2}\left(Y - B\cdot c\right)^T\left(Y - B\cdot c\right) \; + \lambda^T \cdot H\cdot c$$

with λ a vector of **Lagrangian multipliers**.

The constrained optimum is located at the location (c,λ) for which the following partial derivatives are zero:

$$\frac{\partial L(c,\lambda)}{\partial c} = 0, \qquad \frac{\partial L(c,\lambda)}{\partial \lambda} = 0$$

Simplex B-Spline Estimation

Equality constrained OLS estimation

We augment the system into the Lagrangian $L(c,\lambda)$:

$$L(c,\lambda) = \frac{1}{2}\left(Y - B\cdot c\right)^T \left(Y - B\cdot c\right) \; +\lambda^T \cdot H \cdot c$$

The partial derivatives of the Lagrangian are:

$$\frac{\partial L(c,\lambda)}{\partial c} = -\left(Y - B\cdot c\right)^T \cdot B + \lambda^T \cdot H = 0,$$

$$\frac{\partial L(c,\lambda)}{\partial \lambda} = H \cdot c = 0$$

Some reorganizing leaves us with:

$$\frac{\partial L(c,\lambda)}{\partial c} = -B^T \cdot Y + B^T \cdot B\cdot c + H^T \cdot \lambda = 0,$$

$$\frac{\partial L(c,\lambda)}{\partial \lambda} = H \cdot c = 0$$

Simplex B-Spline Estimation

Equality constrained OLS estimation

We can reformulate the partial derivatives as follows

$$B^T \cdot B \cdot c + H^T \cdot \lambda = B^T \cdot Y,$$
$$H \cdot c = 0$$

Which can easily reformulated into the Karush-Kuhn-Tucker (KKT) matrix:

$$\begin{bmatrix} B^T \cdot B & H^T \\ H & 0 \end{bmatrix} \cdot \begin{bmatrix} c \\ \lambda \end{bmatrix} = \begin{bmatrix} B^T \cdot Y \\ 0 \end{bmatrix}$$

The constrained OLS B-coefficient / Lagrangian estimator is:

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} B^T \cdot B & H^T \\ H & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} B^T \cdot Y \\ 0 \end{bmatrix}$$

Simplex B-Spline Estimation

Equality constrained OLS estimation

The OLS B-coefficient / Lagrangian estimator is:

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} B^T \cdot B & H^T \\ H & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} B^T \cdot Y \\ 0 \end{bmatrix}$$

This is equivalent to:

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} \cdot \begin{bmatrix} B^T \cdot Y \\ 0 \end{bmatrix}$$

The constrained OLS B-coefficient estimator is:

$$\hat{c} = C_1 \cdot B^T \cdot Y$$

With statistics:

$$Cov\{\hat{c}\} = C_1, \qquad Var\{\hat{c}\} = diag(C_1)$$

Simplex B-Spline Estimation

Equality constrained OLS estimation

Note that if H is rank-deficient (i.e. $\mathrm{rank}\, H < \mathrm{row}\, H$), the KKT matrix is singular. In that case, we can use the Moore-Penrose pseudo inverse...

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} B^T \cdot B & H^T \\ H & 0 \end{bmatrix}^+ \cdot \begin{bmatrix} B^T \cdot Y \\ 0 \end{bmatrix}$$

However, we may also use a more efficient iterative solver:

First Iteration: $\hat{c}^{(1)} = \left(2B^T \cdot B + \frac{1}{\varepsilon} H^T \cdot H \right)^{-1} \cdot \left(2B^T \cdot Y - H^T \cdot \hat{\lambda}^{(0)} \right), \qquad 0 < \varepsilon \leq 1$

Next Iteration: $\hat{c}^{(k+1)} = \left(2B^T \cdot B + \frac{1}{\varepsilon} H^T \cdot H \right)^{-1} \cdot 2B^T \cdot B \cdot \hat{c}^{(k)}, \qquad 0 < \varepsilon \leq 1$

With ε a small number (e.g. 10^{-6}), and with $\hat{\lambda}^{(0)}$ an initial estimate for the Lagrange multipliers (e.g. $\hat{\lambda}^{(0)} = 1$).

Simplex B-Spline Estimation

MATLAB Demo

Fitting a bivariate (2-D) dataset with a true simplex spline.

Simplex B-Spline Estimation

Elegant Alternative: null-space projection

The continuity conditions $Hc = 0$ imply that our estimated B-coefficients must be in the null-space of H: $\hat{c} \in \mathit{null}(H)$

Let Γ be defined as a basis for the null space of the smoothness constraints:

$$\Gamma = \mathrm{null}(H)$$

Note: this operation can be costly computationally!

Let $\tilde{c}$ be the 'free' (=unconstrained) B-coefficients that are related to the full set of B-coefficients as follows:

$$c = \Gamma \cdot \tilde{c}$$

The least squares problem in terms of real B-coefficients and constraints can therefore be expressed in unconstrained B-coefficients:

$$\min_{c} \left(Y - A \cdot c\right)^{T} \left(Y - A \cdot c\right) \qquad s.t. \quad H \cdot c = 0$$

$$= \min_{\tilde{c}} \left(Y - A \cdot \Gamma \cdot \tilde{c}\right)^{T} \left(Y - A \cdot \Gamma \cdot \tilde{c}\right)$$

Simplex B-Spline Estimation

Elegant Alternative: null-space projection

We have reformulated the constrained LS problem into a unconstrained problem in ‘free’ B-coefficient space:

$$\min_{\tilde{c}} \left(Y - A \cdot \Gamma \cdot \tilde{c}\right)^T \left(Y - A \cdot \Gamma \cdot \tilde{c}\right)$$

We can simplify this by projecting the regression matrix on Γ:

$$\tilde{A} = A \cdot \Gamma$$

The unconstrained LS estimator for the ‘free’ B-coefficients is:

$$\hat{\tilde{c}} = \left(\tilde{A}^T \tilde{A}\right)^{-1} \tilde{A}^T Y$$

The ‘full’ (unconstrained + constrained) B-coefficients are obtained under the reverse projection:

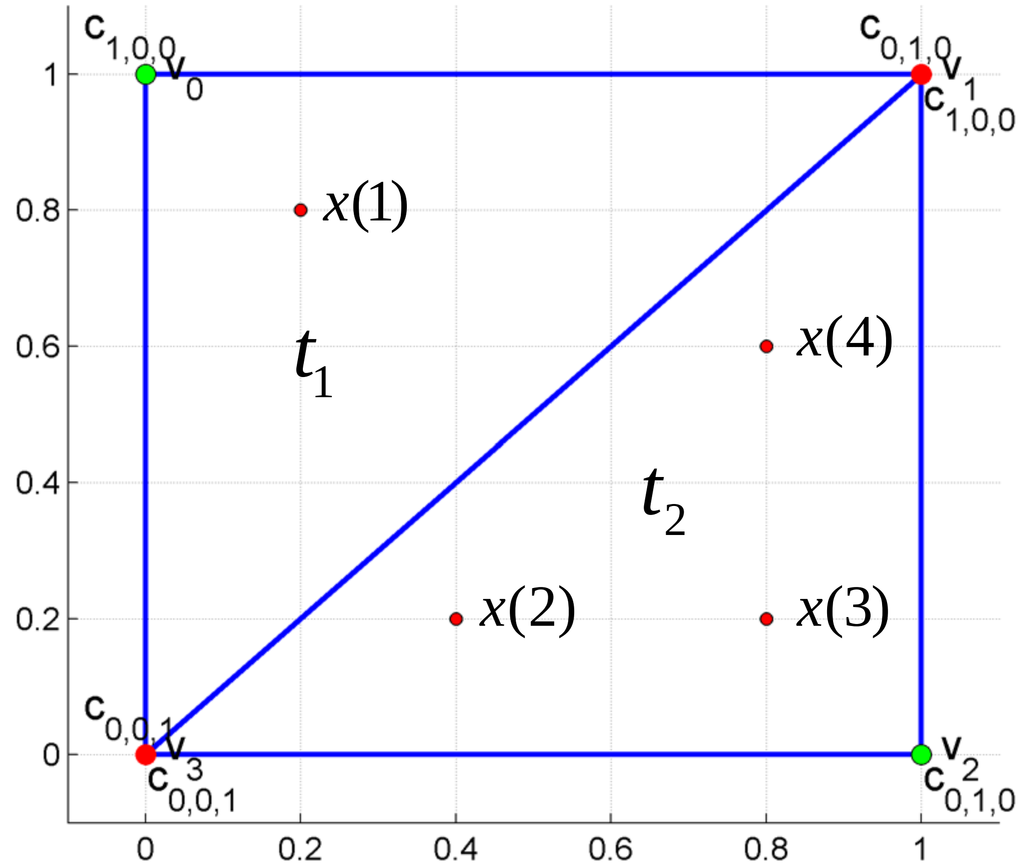
$$\hat{c} = \Gamma \hat{\tilde{c}}$$

Simplex B-Spline Estimation

Example 6.8: Estimating a linear simplex spline with 0st order continuity

Given data:

$$v_0 = (0,1),\ \ v_1 = (1,1),$$
$$v_2 = (1,0),\ v_3 = (0,0),$$
$$t_1 = \langle v_0, v_1, v_3 \rangle,$$
$$t_2 = \langle v_1, v_2, v_3 \rangle,$$
$$x = \begin{bmatrix} 0.2 & 0.8 \\ 0.4 & 0.2 \\ 0.8 & 0.2 \\ 0.8 & 0.6 \end{bmatrix},\quad Y = \begin{bmatrix} -1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$



Simplex B-Spline Estimation

Example 6.8: Estimating a linear simplex spline with 0st order continuity

Data membership search:

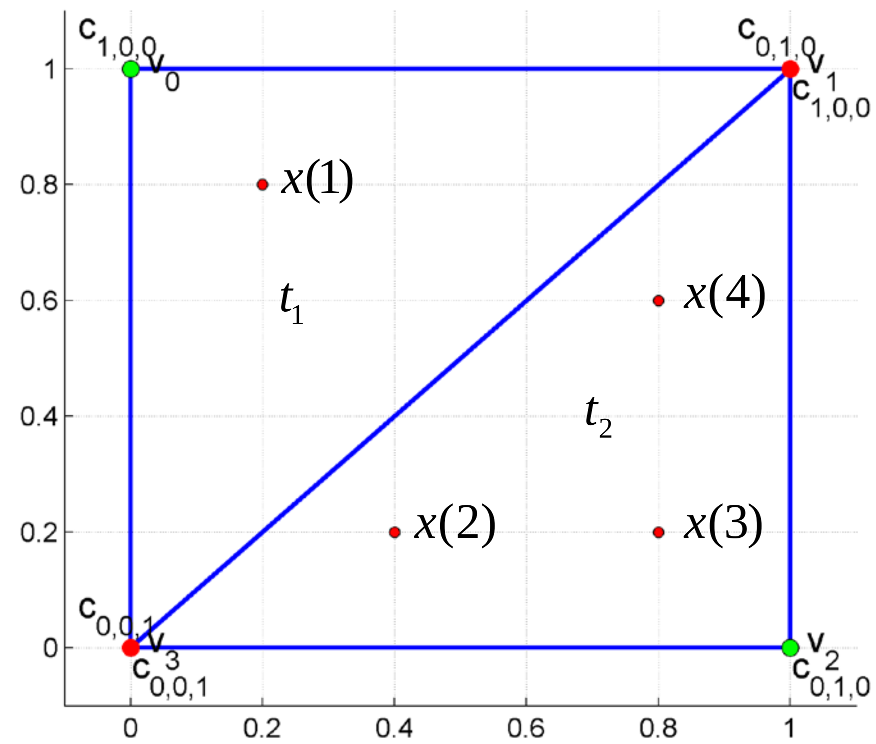
$$x(1) \in t_1$$

$$x(2), x(3), x(4) \in t_2$$

Barycentric transformation matrices:

$$A_{t_1} = \begin{bmatrix} v_{1_x} - v_{0_x} & v_{3_x} - v_{0_x} \\ v_{1_y} - v_{0_y} & v_{3_y} - v_{0_y} \end{bmatrix} =$$

$$A_{t_2} = \begin{bmatrix} v_{2_x} - v_{1_x} & v_{3_x} - v_{1_x} \\ v_{2_y} - v_{1_y} & v_{3_y} - v_{1_y} \end{bmatrix} =$$



Simplex B-Spline Estimation

$$x = \begin{bmatrix} 0.2 & 0.8 \\ 0.4 & 0.2 \\ 0.8 & 0.2 \\ 0.8 & 0.6 \end{bmatrix},$$

$$v_0 = (0,1),\ v_1 = (1,1)$$

Example 6.8: Estimating a linear simplex spline

Barycentric coordinate transformations for simplex t_1:

$$\begin{bmatrix} b_1(1) & b_2(1) \end{bmatrix}^T = A_{t_1}^{-1}(x^T(1) - v_0^T) =$$

Barycentric coordinate transformations for simplex t_2:

$$\begin{bmatrix} b_1(2) & b_2(2) \end{bmatrix}^T = A_{t_2}^{-1}(x^T(2) - v_1^T) =$$

$$\begin{bmatrix} b_1(3) & b_2(3) \end{bmatrix}^T = A_{t_2}^{-1}(x^T(3) - v_1^T) =$$

$$\begin{bmatrix} b_1(4) & b_2(4) \end{bmatrix}^T = A_{t_2}^{-1}(x^T(4) - v_1^T) =$$

Simplex B-Spline Estimation

Example 6.8: Estimating a linear simplex spline with 0st order continuity

The vector B-form in this case (d=1) is:

$$p(x) = B^1(b_{t_j}(x)) \cdot c^{t_j} = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} \cdot \begin{bmatrix} c^{t_j}_{1,0,0} & c^{t_j}_{0,1,0} & c^{t_j}_{0,0,1} \end{bmatrix}^T$$

The global regression matrix is constructed as follows: $B = \begin{bmatrix} B_{t_1} & 0 \\ 0 & B_{t_2} \end{bmatrix}$

$B_{t_1} =$ $\qquad\qquad$ $B_{t_2} =$

The global regression matrix is:

$B =$

$$b(x(1)) = \begin{bmatrix} 0.6 & 0.2 & 0.2 \end{bmatrix}$$
$$b(x(2)) = \begin{bmatrix} 0.2 & 0.2 & 0.6 \end{bmatrix}$$
$$b(x(3)) = \begin{bmatrix} 0.2 & 0.6 & 0.2 \end{bmatrix}$$
$$b(x(4)) = \begin{bmatrix} 0.6 & 0.2 & 0.2 \end{bmatrix}$$

Simplex B-Spline Estimation

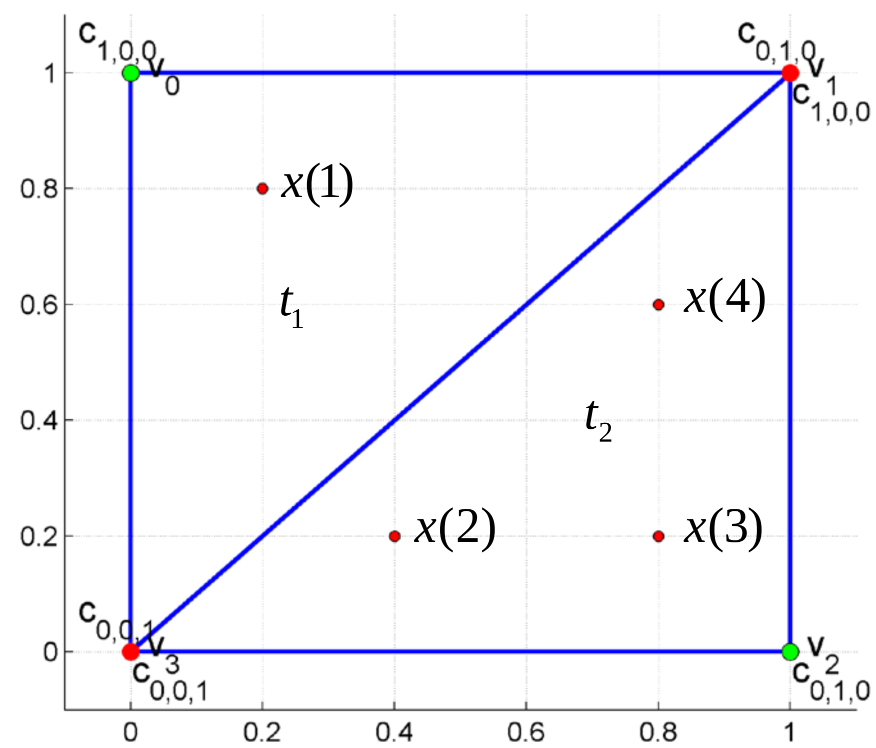
Example 6.8: Estimating a linear simplex spline with 0st order continuity

Continuity conditions (m=0):

$$c_{0,\kappa_1,\kappa_2}^{t_1} = \sum_{|\gamma|=0} c_{(\kappa_0,0,\kappa_2)+\gamma}^{t_2} B_\gamma^0(v_2)$$

$$= c_{(\kappa_0,0,\kappa_2)+\gamma}^{t_2}$$

Continuity conditions are:



Smoothness matrix H:

$$H =$$

Simplex B-Spline Estimation

Example 6.8: Estimating a linear simplex spline with 0st order continuity

Construct the KKT matrix:

$$M = \begin{bmatrix} B^T \cdot B & H^T \\ H & 0 \end{bmatrix} =$$

Simplex B-Spline Estimation

Example 6.8: Estimating a linear simplex spline with 0st order continuity

Construct the vector $B^T Y$:

$$B^T \cdot Y = \begin{bmatrix} 0.6 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0.2 & 0.6 \\ 0 & 0 & 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0 & 0.6 & 0.2 & 0.2 \end{bmatrix}^T \cdot \begin{bmatrix} -1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.6 \\ -0.2 \\ -0.2 \\ 1.4 \\ 1.8 \\ 1.8 \end{bmatrix}$$

Simplex B-Spline Estimation

Example 6.8: Estimating a linear simplex spline with 0st order continuity

Estimate B-coefficients & Lagrangian multipliers:

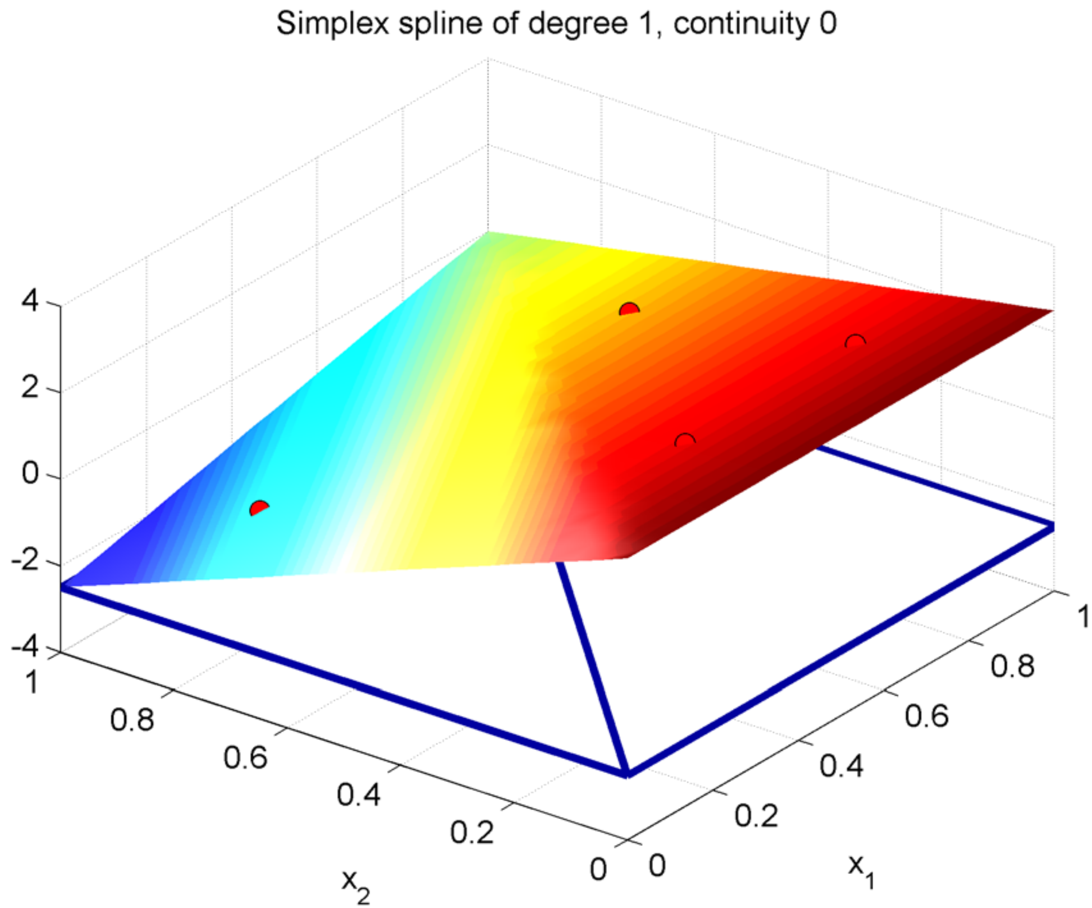
$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} 0.36 & 0.12 & 0.12 & 0 & 0 & 0 & 0 & 0 \\ 0.12 & 0.04 & 0.04 & 0 & 0 & 0 & -1 & 0 \\ 0.12 & 0.04 & 0.04 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0.44 & 0.28 & 0.28 & 1 & 0 \\ 0 & 0 & 0 & 0.28 & 0.44 & 0.28 & 0 & 0 \\ 0 & 0 & 0 & 0.28 & 0.28 & 0.44 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} -0.6 \\ -0.2 \\ -0.2 \\ 1.4 \\ 1.8 \\ 1.8 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -2.5 \\ 0 \\ 2.5 \\ 0 \\ 2.5 \\ 2.5 \\ 0 \\ 0 \end{bmatrix}$$

Simplex B-Spline Estimation

Example 6.8: Estimating a linear simplex spline with 0st order continuity

Simplex B-spline model is:

$$p(x) = B^1(b_{t_j}(x)) \cdot \begin{bmatrix} -2.5 \\ 0 \\ 2.5 \\ 0 \\ 2.5 \\ 2.5 \end{bmatrix}, \quad x \in t_j$$



Demonstration: F-16 aerodynamic model

Original wind tunnel based model (5-D):

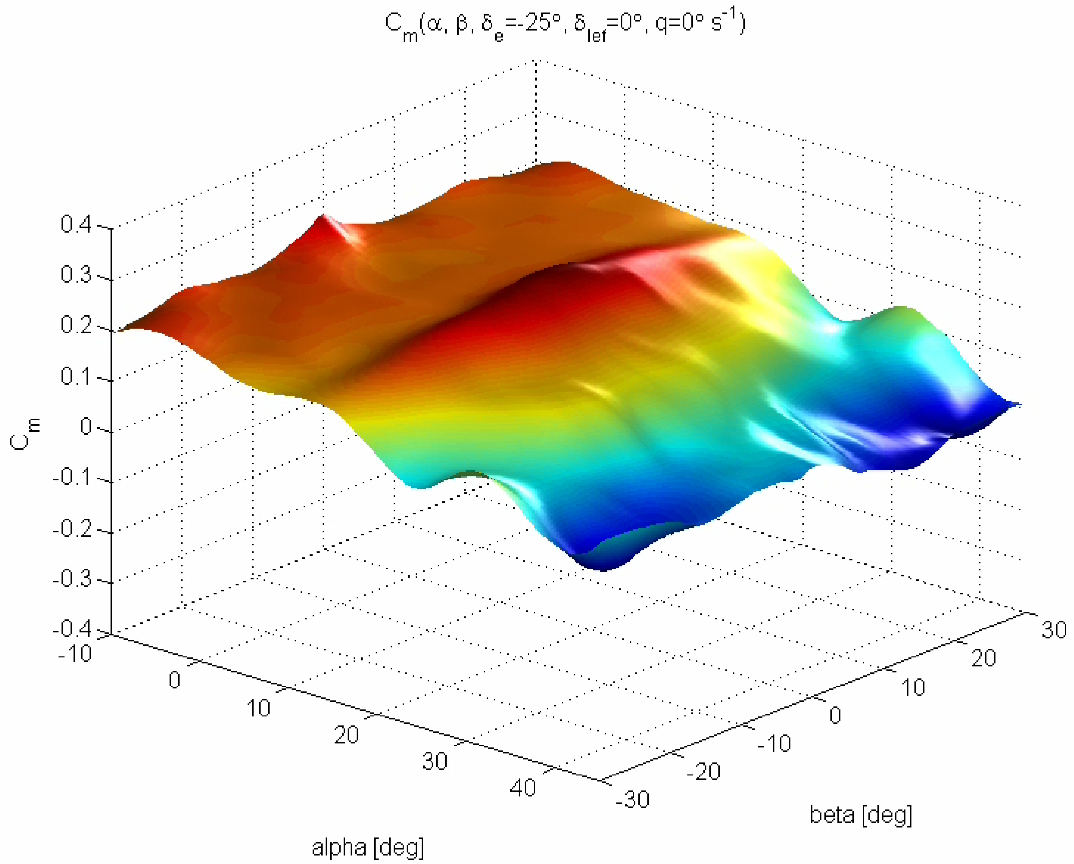
- 48 data tables (3-D, 2-D, and 1-D)
- Linearly interpolated
- Total size: 15 Mb (1.8M datapoints)
- No continuity between table grid cells
- Limited resolution
- Linear interpolation between grid cells

Simplex spline based model (5-D):

- 36 spline functions (3-D, 2-D, and 1-D)
- Up to 6th degree spline polynomials
- Total size: 0.6 Mb (100K B-coefficients & vertices)
- 1st order continuity between spline pieces
- Model error RMS w.r.t. wind tunnel model < 0.5%

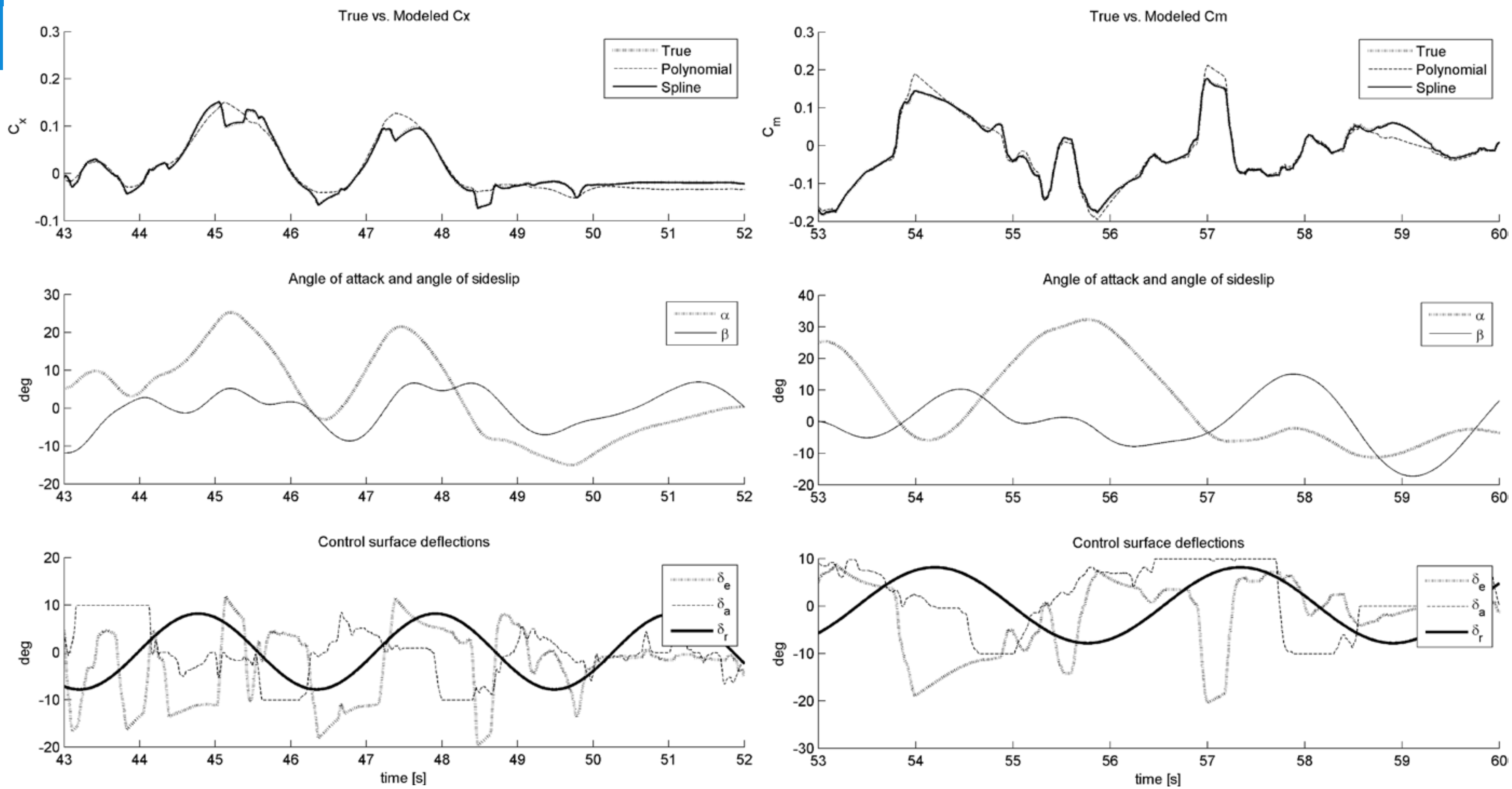
Demonstration: F-16 aerodynamic model

Complete model = linear combination of 3-D, 2-D and 1-D splines:

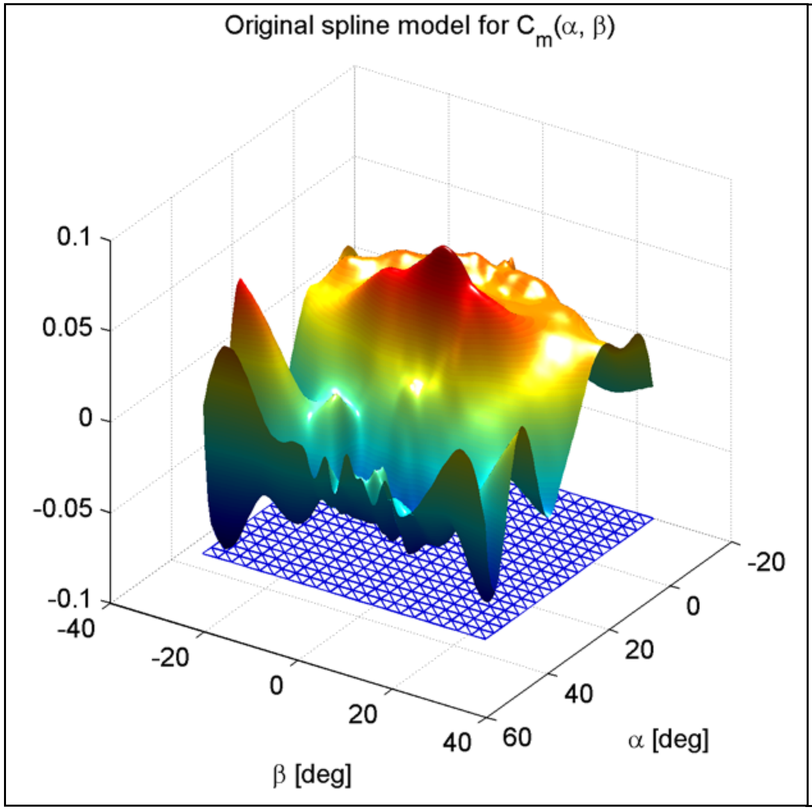


Demonstration: F-16 aerodynamic model

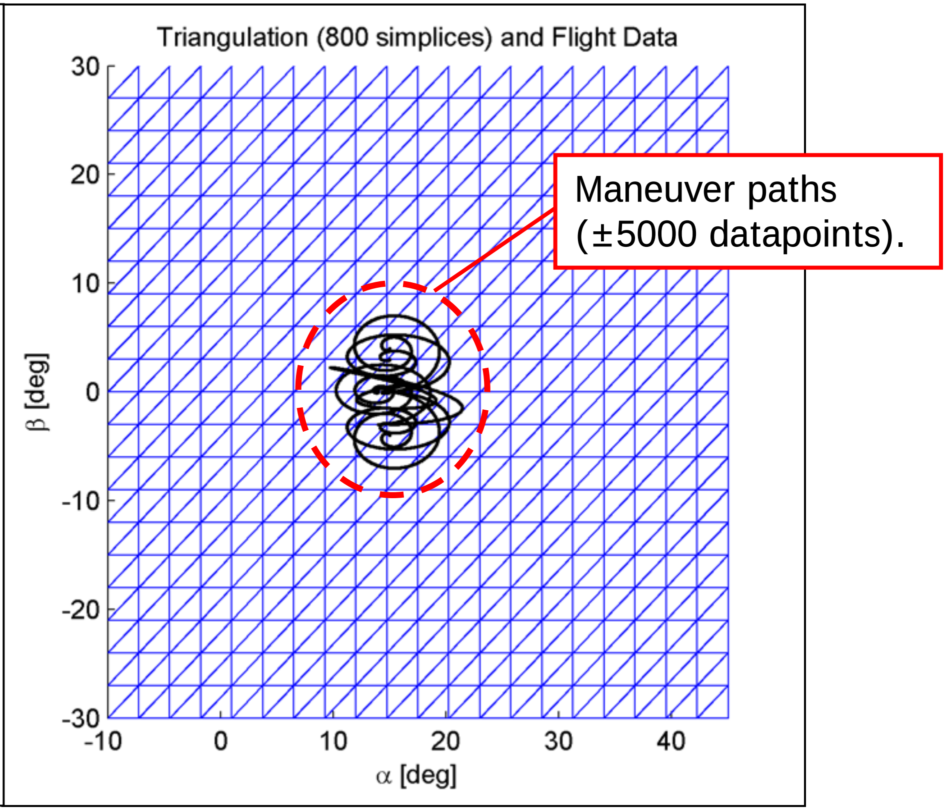
Simulation output from Simplex Spline based aerodynamic model



Demonstration: Local model updates



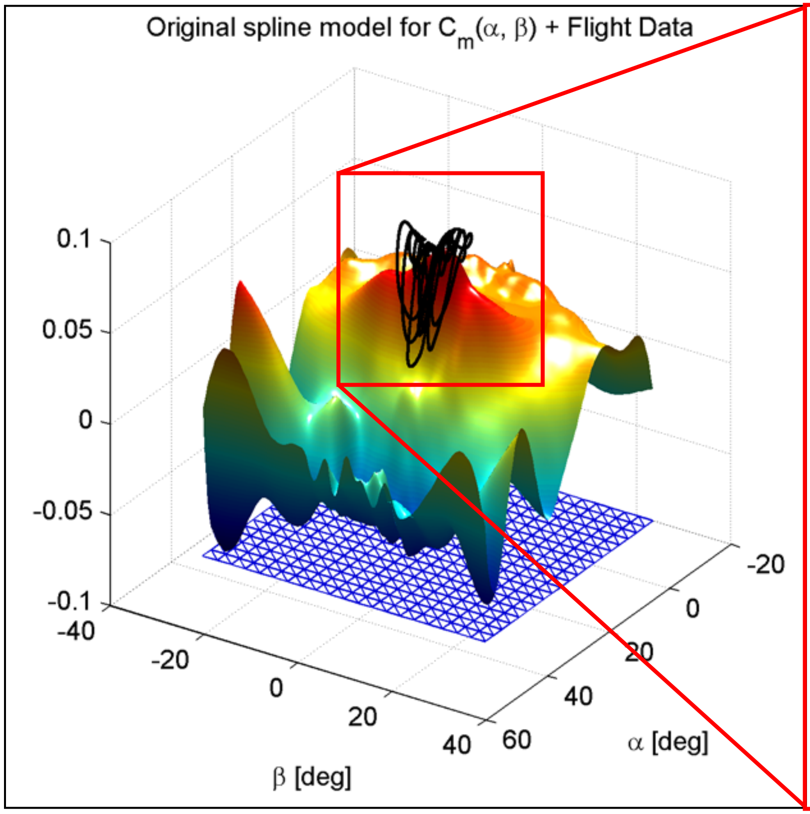
Initial spline model



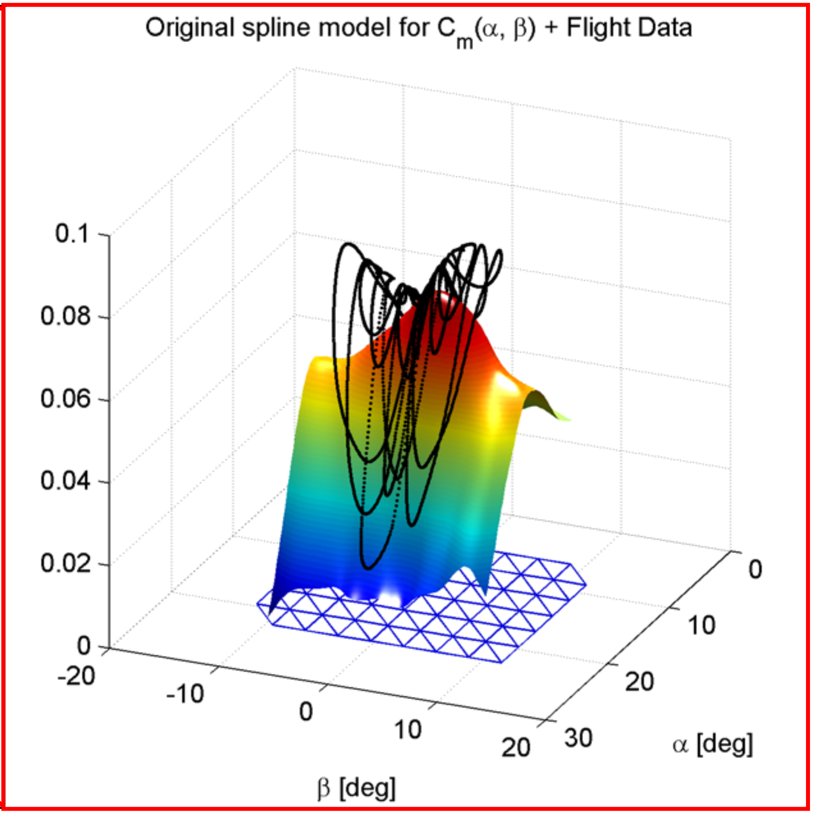
Triangulation and maneuver paths

Demonstration: Local model updates

Step 1: detect error between model and flight test data



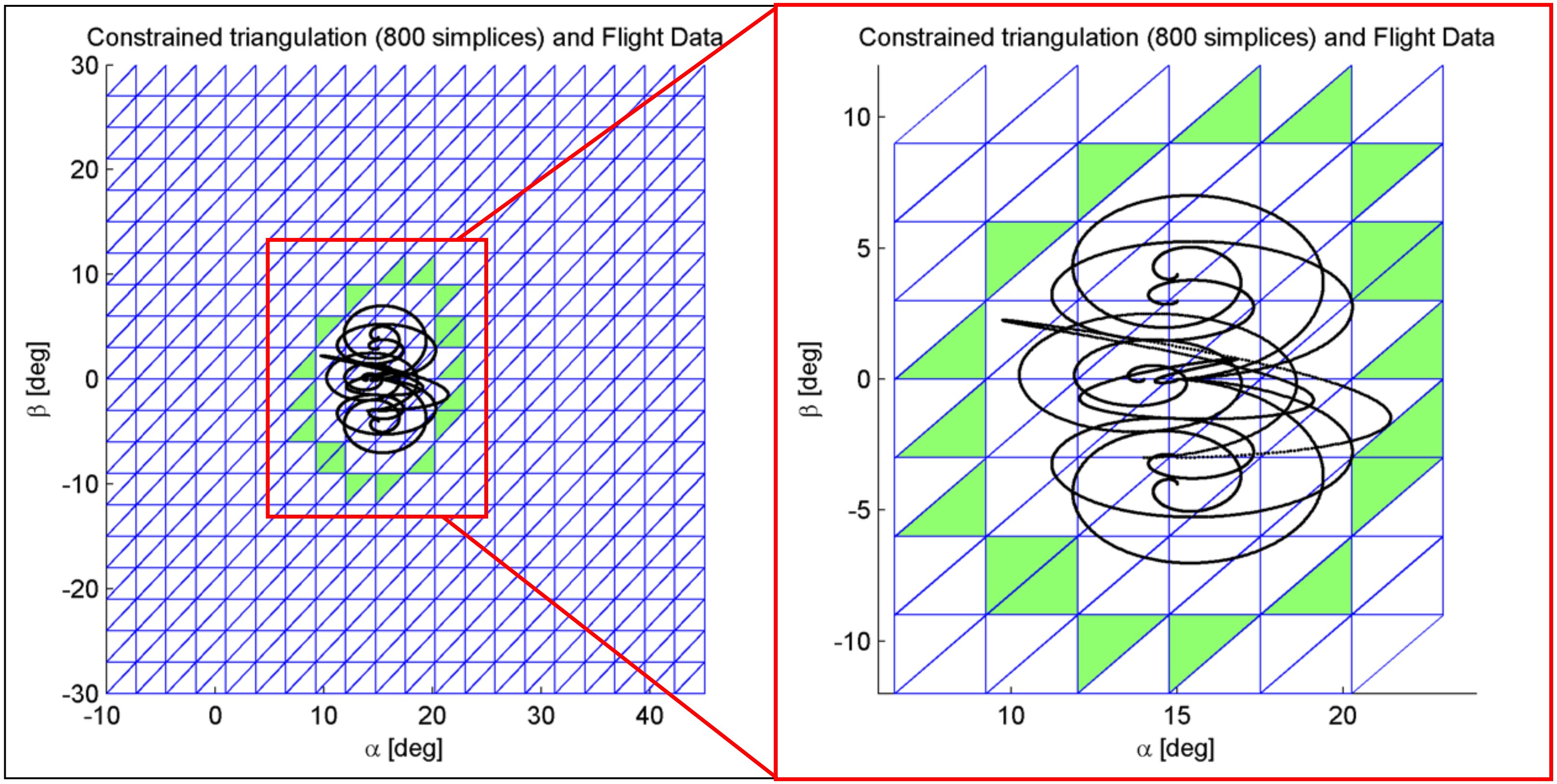
Initial spline model + Flight Data



Detailed view

Demonstration: Local model updates

Step 2: Constrain all model parameters outside affected model region

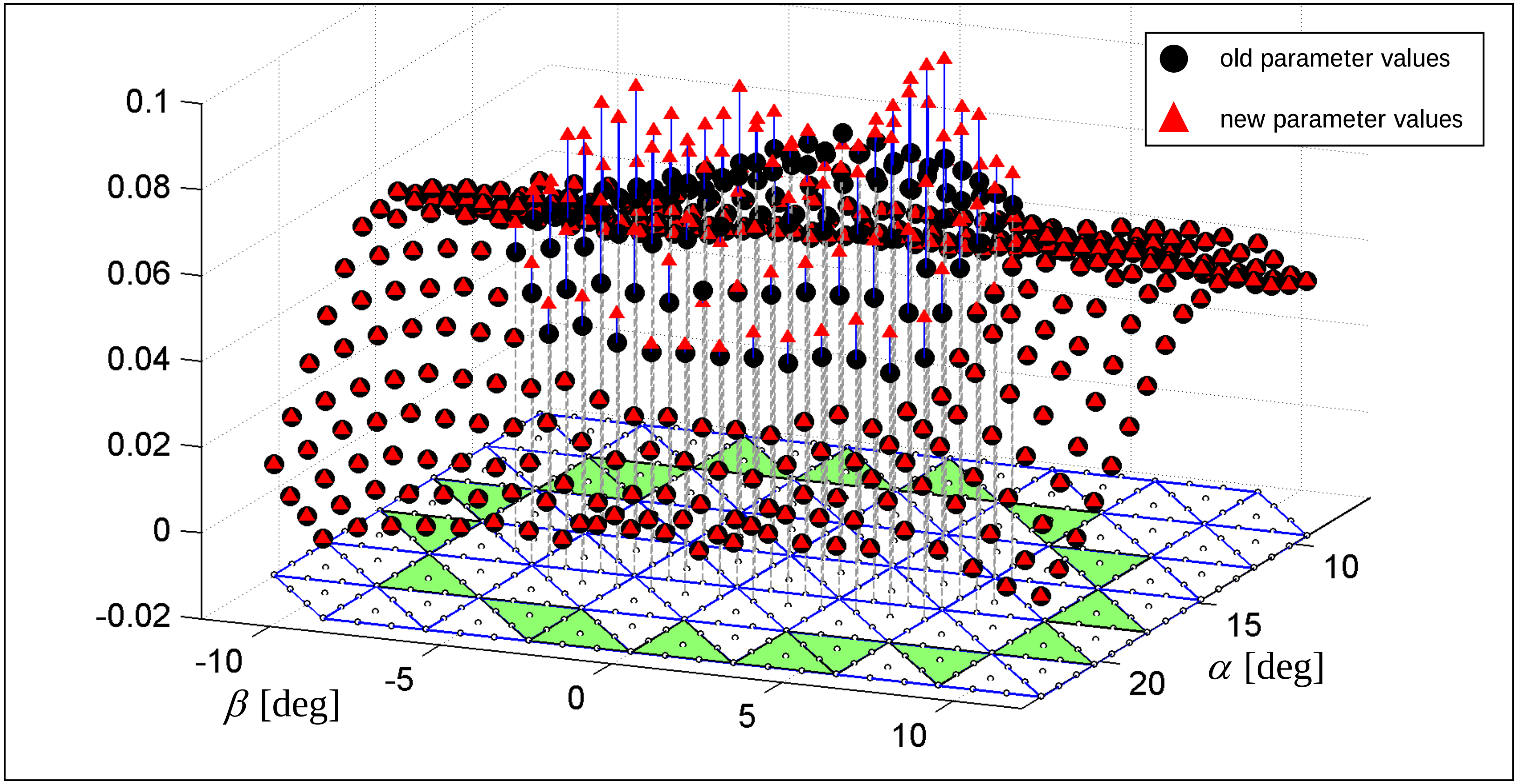


Constrained parameters (green triangles)

Detailed view

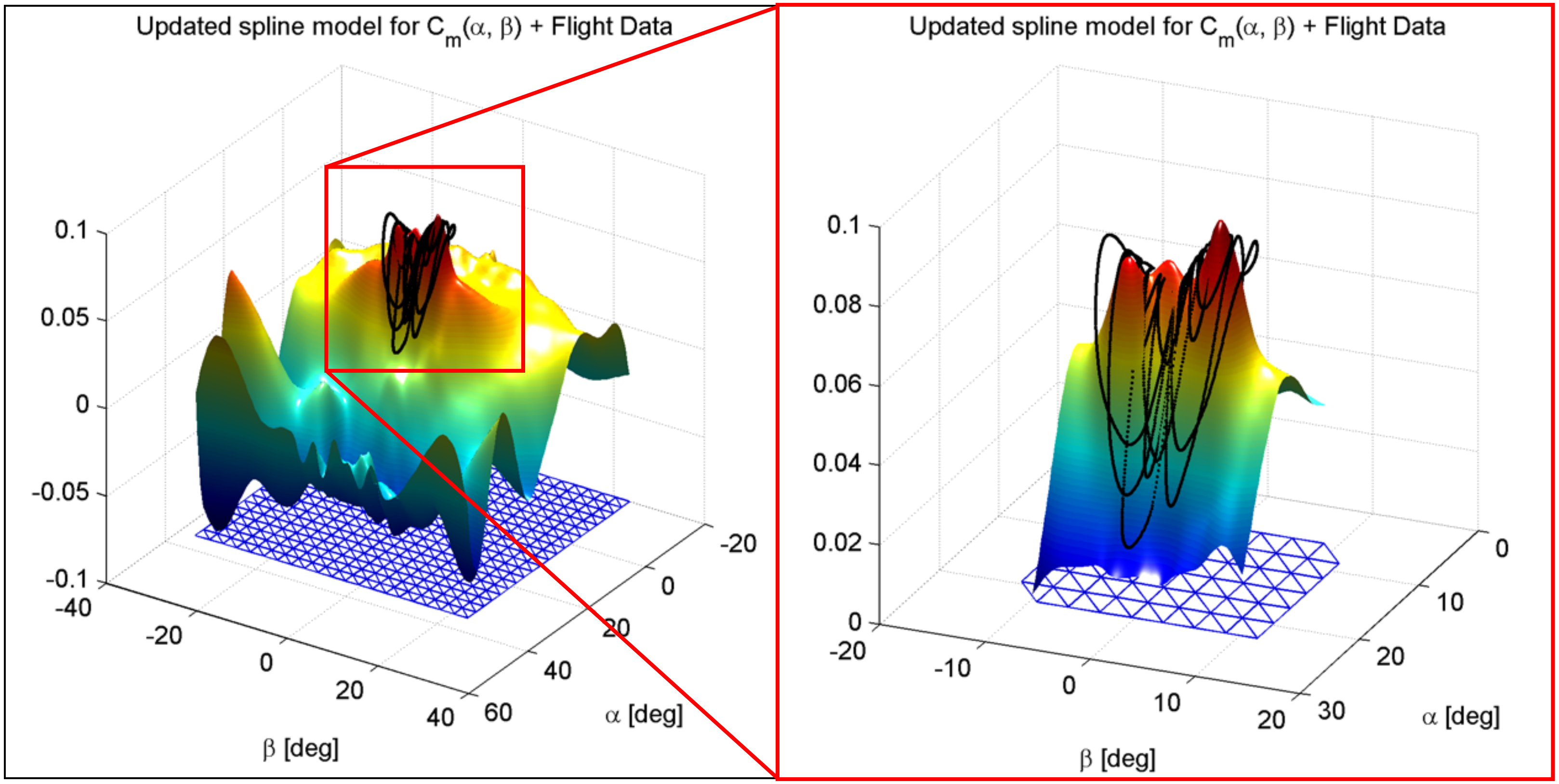
Demonstration: Local model updates

Step 3: Re-estimate model parameters inside affected model region



Demonstration: Local model updates

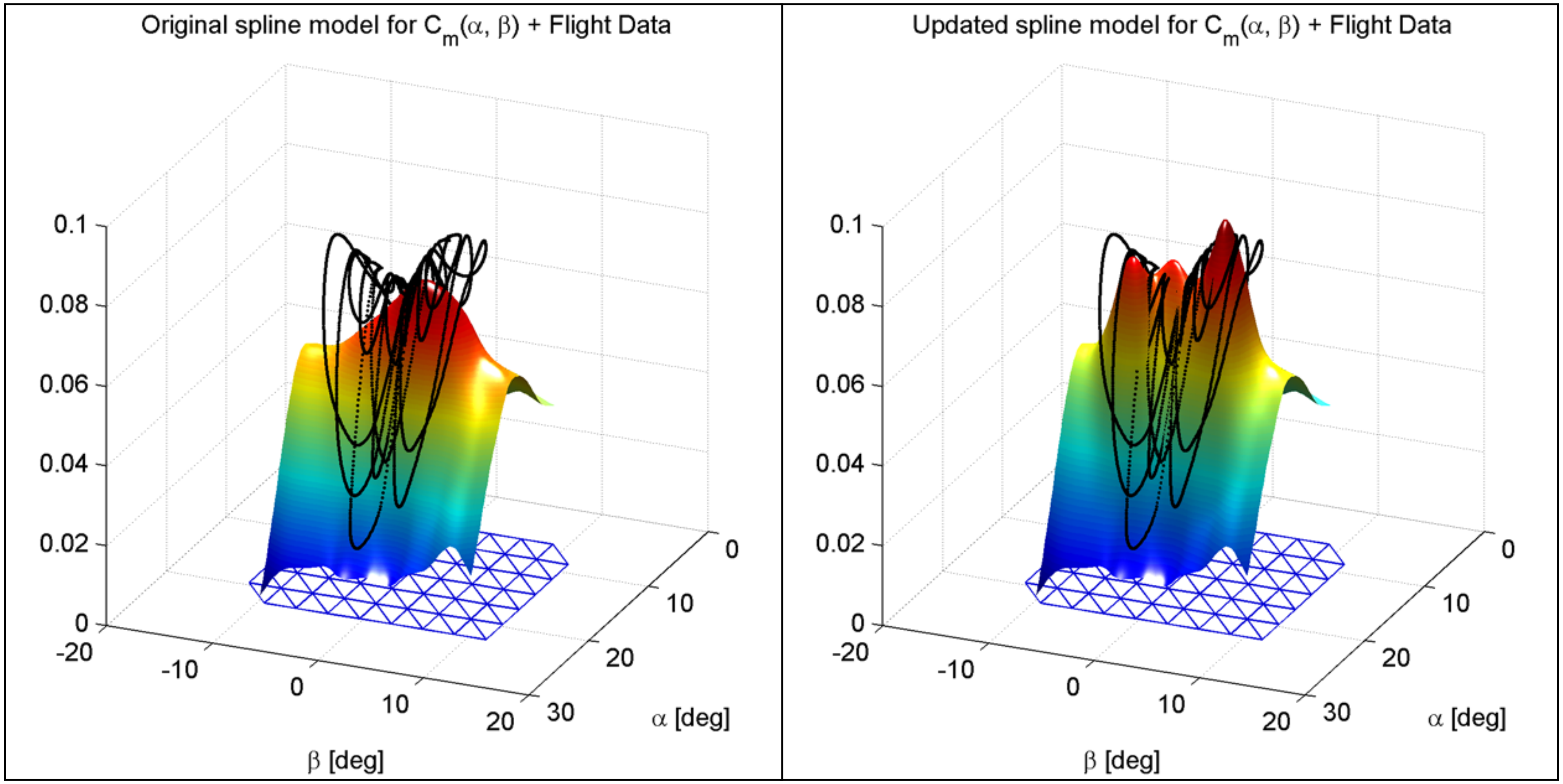
Step 4: Re-evaluate updated model



Updated spline model

Detailed view

Demonstration: Local model updates



Original spline model (detail) | Updated spline model (detail)

Summary

Comparison of general function approximators

- Neural Networks
  - \+ A general nonlinear function approximator
  - \+ Can fit any scattered nonlinear data set
  - \- Global basis functions: non-sparse solution systems
  - \- Training of NN presents a nonlinear optimization problem
- Multivariate Simplex B-Splines
  - \+ A general function approximator compatible with **linear regression** methods
  - \+ Can fit any scattered nonlinear data set
  - \+ **Local basis functions: efficient sparse solution systems**
  - \+ / - Requires underlying geometric support structure called a 'triangulation'

Advanced Applications

Advanced Applications of Multivariate Simplex Splines:

❖ Physical splines

- Use basis transformations to transform spline polynomials from Barycentric coordinate space into physical space.

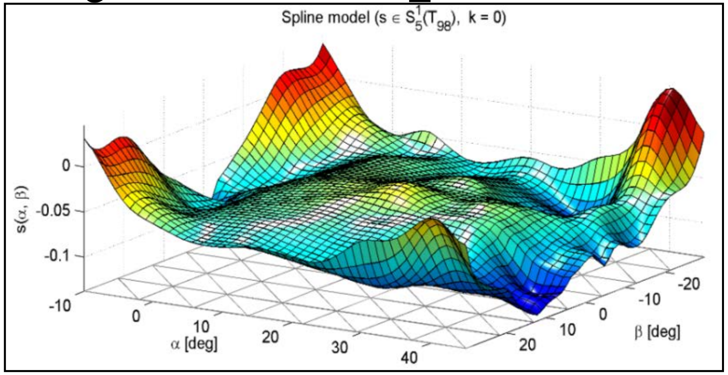
❖ Adaptive simplex splines

- Use online (recursive) identification techniques to (locally) adapt a simplex spline model after changes to the system dynamics.

❖ Solve and control partial differential equations using simplex splines.

- Solve PDE's (wave equation, heat equation) using simplex splines.

Advanced Applications: Physical Splines



So, what does it all mean? Can we somehow interpret the splines in physical terms?

Remember that on every simplex we have B-form polynomial in barycentric coordinates:

$$p(x) = B^d(b_{t_j}(x)) \cdot c^{t_j} \qquad \forall x \in t_j$$

$$= \begin{bmatrix} B^d_{d,0,\dots,0}(b_{t_j}(x)) & B^d_{d-1,1,\dots,0}(b_{t_j}(x)) & \cdots & B^d_{0,0,\dots,d}(b_{t_j}(x)) \end{bmatrix} \cdot \begin{bmatrix} c^{t_j}_{d,0,\dots,0} \\ c^{t_j}_{d-1,1,\dots,0} \\ \vdots \\ c^{t_j}_{0,0,\dots,d} \end{bmatrix}$$

Example (2-dimensional case, degree = 3):

$$p(x) = B^3(b_{t_j}(x,y)) \cdot c^{t_j} = [b_0^3 \;\; 3b_0^2 b_1^1 \;\; 3b_0^2 b_2^1 \;\; 3b_0^1 b_1^2 \;\; 6b_0^1 b_1^1 b_2^1 \;\; 3b_0^1 b_2^2 \;\; b_1^3 \;\; 3b_1^2 b_2^1 \;\; 3b_1^1 b_2^2 \;\; b_2^3] \cdot \mathrm{c}$$

Advanced Applications: Physical Splines

...In most cases a simple polynomial model structure is used for aerodynamic model identification...

Example (right):
the aerodynamic model of El-Al Boeing 747-200 (*).

$$
\begin{aligned}
C_X &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\alpha^2}}\alpha^2 + C_{X_q}\frac{q\bar{c}}{V} + C_{X_{\delta_{e_{ir}}}}|\delta_{e_{ir}}| + C_{X_{\delta_{e_{il}}}}|\delta_{e_{il}}| + C_{X_{\delta_{e_{or}}}}|\delta_{e_{or}}| \\
&\quad + C_{X_{\delta_{e_{ol}}}}|\delta_{e_{ol}}| + C_{X_{i_h}}|i_h| + C_{X_{\delta_{sp1}}}\delta_{sp1} + \ldots + C_{X_{\delta_{sp12}}}\delta_{sp12} + C_{X_{\delta_{f_o}}}\delta_{f_o} + C_{X_{\delta_{f_i}}}\delta_{f_i} \\
&\quad + C_{X_{EPR_1}}EPR_1 + \ldots + C_{X_{EPR4}}EPR_4 + \boxed{C_{X_\beta}\beta + C_{X_p}\frac{pb}{2V} + C_{X_r}\frac{rb}{2V}} \qquad (4.67)
\end{aligned}
$$

$$
\begin{aligned}
C_Z &= C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_q}\frac{q\bar{c}}{V} + C_{Z_{\delta_{e_{ir}}}}\delta_{e_{ir}} + C_{Z_{\delta_{e_{il}}}}\delta_{e_{il}} + C_{Z_{\delta_{e_{or}}}}\delta_{e_{or}} + C_{Z_{\delta_{e_{ol}}}}\delta_{e_{ol}} + \\
&\quad + C_{Z_{i_h}}i_h + C_{Z_{\delta_{sp1}}}\delta_{sp1} + \ldots + C_{Z_{\delta_{sp12}}}\delta_{sp12} + C_{Z_{\delta_{f_o}}}\delta_{f_o} + C_{Z_{\delta_{f_i}}}\delta_{f_i} \\
&\quad + C_{Z_{EPR_1}}EPR_1 + \ldots + C_{Z_{EPR4}}EPR_4 + \boxed{C_{Z_\beta}\beta + C_{Z_p}\frac{pb}{2V} + C_{Z_r}\frac{rb}{2V}} \qquad (4.68)
\end{aligned}
$$

$$
\begin{aligned}
C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{q\bar{c}}{V} + C_{m_{\delta_{e_{ir}}}}\delta_{e_{ir}} + C_{m_{\delta_{e_{il}}}}\delta_{e_{il}} + C_{m_{\delta_{e_{or}}}}\delta_{e_{or}} + C_{m_{\delta_{e_{ol}}}}\delta_{e_{ol}} + \\
&\quad + C_{m_{i_h}}i_h + C_{m_{\delta_{sp1}}}\delta_{sp1} + \ldots + C_{m_{\delta_{sp12}}}\delta_{sp12} + C_{m_{\delta_{f_o}}}\delta_{f_o} + C_{m_{\delta_{f_i}}}\delta_{f_i} \\
&\quad + C_{m_{EPR_1}}EPR_1 + \ldots + C_{m_{EPR4}}EPR_4 + \boxed{C_{m_\beta}\beta + C_{m_p}\frac{pb}{2V} + C_{m_r}\frac{rb}{2V}} \qquad (4.69)
\end{aligned}
$$

$$
\begin{aligned}
C_Y &= C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{pb}{2V} + C_{Y_r}\frac{rb}{2V} + C_{Y_{\delta_{a_{ir}}}}\delta_{a_{ir}} + C_{Y_{\delta_{a_{il}}}}\delta_{a_{il}} + C_{Y_{\delta_{a_{or}}}}\delta_{a_{or}} \\
&\quad + C_{Y_{\delta_{a_{ol}}}}\delta_{a_{ol}} + C_{Y_{\delta_{r_u}}}\delta_{r_u} + C_{Y_{\delta_{r_l}}}\delta_{r_l} + C_{Y_{\delta_{sp1}}}\delta_{sp1} + \ldots + C_{Y_{\delta_{sp12}}}\delta_{sp12} \\
&\quad + \boxed{C_{Y_\alpha}\alpha + C_{Y_q}\frac{q\bar{c}}{V}} + \boxed{C_{Y_{EPR_1}}EPR_1 + \ldots + C_{Y_{EPR4}}EPR_4} \qquad (4.70)
\end{aligned}
$$

$$
\begin{aligned}
C_l &= C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l_{\delta_{a_{ir}}}}\delta_{a_{ir}} + C_{l_{\delta_{a_{il}}}}\delta_{a_{il}} + C_{l_{\delta_{a_{or}}}}\delta_{a_{or}} + C_{l_{\delta_{a_{ol}}}}\delta_{a_{ol}} + \\
&\quad + C_{l_{\delta_{r_u}}}\delta_{r_u} + C_{l_{\delta_{r_l}}}\delta_{r_l} + C_{l_{\delta_{sp1}}}\delta_{sp1} + \ldots + C_{l_{\delta_{sp12}}}\delta_{sp12} + \boxed{C_{l_\alpha}\alpha + C_{l_q}\frac{q\bar{c}}{V}} + \\
&\quad + \boxed{C_{l_{EPR_1}}EPR_1 + \ldots + C_{l_{EPR4}}EPR_4} \qquad (4.71)
\end{aligned}
$$

$$
\begin{aligned}
C_n &= C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V} + C_{n_r}\frac{rb}{2V} + C_{n_{\delta_{a_{ir}}}}\delta_{a_{ir}} + C_{n_{\delta_{a_{il}}}}\delta_{a_{il}} + C_{n_{\delta_{a_{or}}}}\delta_{a_{or}} \\
&\quad + C_{n_{\delta_{a_{ol}}}}\delta_{a_{ol}} + C_{n_{\delta_{r_u}}}\delta_{r_u} + C_{n_{\delta_{r_l}}}\delta_{r_l} + C_{n_{\delta_{sp1}}}\delta_{sp1} + \ldots + C_{n_{\delta_{sp12}}}\delta_{sp12} \\
&\quad + \boxed{C_{n_\alpha}\alpha + C_{n_q}\frac{q\bar{c}}{V}} + \boxed{C_{n_{EPR_1}}EPR_1 + \ldots + C_{n_{EPR4}}EPR_4} \qquad (4.72)
\end{aligned}
$$

(*) T.J.J. Lombaerts, PhD Thesis, TU-Delft, 2010

Advanced Applications: Physical Splines

For aerodynamic model identification, we often use polynomial model structures for the aerodynamic forces and moments, for example:

$$C_X(\alpha,\delta_e) = C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\alpha^2}}\alpha^2 + C_{X_{\delta_e}}\delta_e + C_{X_{\alpha\delta_e}}\alpha\delta_e$$

Is there some way of relating these (well proven) polynomials to multivariate spline polynomials in barycentric coordinates such that:

$$\begin{aligned} C_X(\alpha,\delta_e) &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\alpha^2}}\alpha^2 + C_{X_{\delta_e}}\delta_e + C_{X_{\alpha\delta_e}}\alpha\delta_e \\ &= p(b(\alpha,\delta_e)) \end{aligned}$$

Remember the words of *de Boor*:

“Any polynomial can be expressed in the B-form”...

Advanced Applications: Physical Splines

Let's start looking at a simple (linear) example:

$$
\begin{aligned}
C_X(\alpha,\delta_e) &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\delta_e}}\delta_e \\
&= p(b(\alpha,\delta_e)) \\
&= \sum_{|k|=1} c_k B_k^d(b(\alpha,\delta_e)) \\
&= c_{100}B_{100}^1(b(\alpha,\delta_e)) + c_{010}B_{010}^1(b(\alpha,\delta_e)) + c_{001}B_{001}^1(b(\alpha,\delta_e)) \\
&= c_{100}b_{100} + c_{010}b_{010} + c_{001}b_{001}
\end{aligned}
$$

The question now is how to relate the barycentric coordinates $b_{100}, b_{010}, b_{001}$ to their physical counterparts α, δ_e…

We use an alternative expression for the barycentric coordinate transformation:

$$
b(\alpha,\delta_e) = A_t \cdot \begin{bmatrix} \alpha \\ \delta_e \end{bmatrix} + k
$$

see [*] for a proof!

[*] H.J. Tol, C.C. de Visser, E. van Kampen and Q.P. Chu, Nonlinear Multivariate Spline Based Control Allocation for High Performance Aircraft, *AIAA Journal of Guidance, Control, and Dynamics*, 37:1840-1862, **2014**.

Advanced Applications: Physical Splines

We then find the following relationship between the barycentric and the physical coordinates on the simplex t:

$$\begin{bmatrix} b_{100} \\ b_{010} \\ b_{001} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \delta_e \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}$$

see [*] for a proof!

Depends on simplex geometry!

Working out the equations, we get:

$$b_{100} = a_{11}\alpha + a_{12}\delta_e + k_1$$
$$b_{010} = a_{21}\alpha + a_{22}\delta_e + k_2$$
$$b_{001} = a_{31}\alpha + a_{32}\delta_e + k_3$$

[*] H.J. Tol, C.C. de Visser, E. van Kampen and Q.P. Chu, Nonlinear Multivariate Spline Based Control Allocation for High Performance Aircraft, *AIAA Journal of Guidance, Control, and Dynamics*, 37:1840-1862, **2014**.

Advanced Applications: Physical Splines

We then find the following relationship between the barycentric and the physical coordinates:

$$
\begin{aligned}
C_X(\alpha,\delta_e) &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\delta_e}}\delta_e \\
&= c_{100}b_{100} + c_{010}b_{010} + c_{001}b_{001} \\
&= c_{100}(a_{11}\alpha + a_{12}\delta_e + k_1) + c_{010}(a_{21}\alpha + a_{22}\delta_e + k_2) + c_{001}(a_{31}\alpha + a_{32}\delta_e + k_3)
\end{aligned}
$$

Grouping terms:

$$
\begin{aligned}
C_X(\alpha,\delta_e) &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\delta_e}}\delta_e \\
&= (c_{100}k_1 + c_{010}k_2 + c_{001}k_3) + \\
&\quad (c_{100}a_{11}\alpha + c_{010}a_{21}\alpha + c_{001}a_{31}\alpha) + \\
&\quad (c_{100}a_{12}\delta_e + c_{010}a_{22}\delta_e + c_{001}a_{32}\delta_e) \\
&= (c_{100}k_1 + c_{010}k_2 + c_{001}k_3) + (c_{100}a_{11} + c_{010}a_{21} + c_{001}a_{31})\alpha + (c_{100}a_{12} + c_{010}a_{22} + c_{001}a_{32})\delta_e
\end{aligned}
$$

Advanced Applications: Physical Splines

Notice the pattern:

$$C_X(\alpha,\delta_e) = C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\delta_e}}\delta_e$$

$$= \underbrace{(c_{100}k_1 + c_{010}k_2 + c_{001}k_3)}_{C_{X_0}} + \underbrace{(c_{100}a_{11} + c_{010}a_{21} + c_{001}a_{31})}_{C_{X_\alpha}}\alpha + \underbrace{(c_{100}a_{12} + c_{010}a_{22} + c_{001}a_{32})}_{C_{X_{\delta_e}}}\delta_e$$

We found a translation between physical coefficients and B-coefficients!!!

$$\begin{bmatrix} C_{X_0} \\ C_{X_\alpha} \\ C_{X_{\delta_e}} \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 \\ a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix} \begin{bmatrix} c_{100} \\ c_{010} \\ c_{001} \end{bmatrix}$$

$$= \Lambda_t \cdot \begin{bmatrix} c_{100} \\ c_{010} \\ c_{001} \end{bmatrix}$$

Advanced Applications: Physical Splines

Because Λ is invertible, we can easily translate from B-space to Physical space (P-space) and vice versa!

$$\begin{bmatrix} C_{X_0} \\ C_{X_\alpha} \\ C_{X_{\delta_e}} \end{bmatrix} = \Lambda_t \cdot \begin{bmatrix} c_{100} \\ c_{010} \\ c_{001} \end{bmatrix} \rightarrow \Lambda_t^{-1} \begin{bmatrix} C_{X_0} \\ C_{X_\alpha} \\ C_{X_{\delta_e}} \end{bmatrix} = \begin{bmatrix} c_{100} \\ c_{010} \\ c_{001} \end{bmatrix}$$

We can define a transformation matrix Λ for every simplex, and use it to convert any B-spline function into a Physical-space spline (P-spline) function:

$$\mathbf{c}_p = \begin{bmatrix} \Lambda_{t_1} & 0 & 0 & 0 \\ 0 & \Lambda_{t_2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \Lambda_{t_J} \end{bmatrix} \cdot \mathbf{c}$$

Advanced Applications: Physical Splines

Null-space projection based estimator for P-coefficients...

The continuity conditions in terms of physical coefficients c_P are:

$$H \cdot c = H \cdot \Lambda^{-1} \cdot c_P$$

Let Γ_P be defined as $\Gamma_P = \mathit{null}(H\Lambda^{-1})$

We can now project the 'physicalized' regression matrix on Γ_P:

$$\tilde{A}_P = \left(A \cdot \Lambda^{-1}\right) \cdot \Gamma_P$$

The unconstrained LS estimator for the 'smooth' P-coefficients is simply:

$$\hat{\tilde{c}}_P = \left(\tilde{A}_P^{\,T} \tilde{A}_P\right)^{-1} \tilde{A}_P^{\,T} Y$$

Projecting back to 'real' P-coefficient space using Γ_P:

$$\hat{c}_P = \Gamma_P \hat{\tilde{c}}_P$$

The resulting spline function in physical coefficients is:

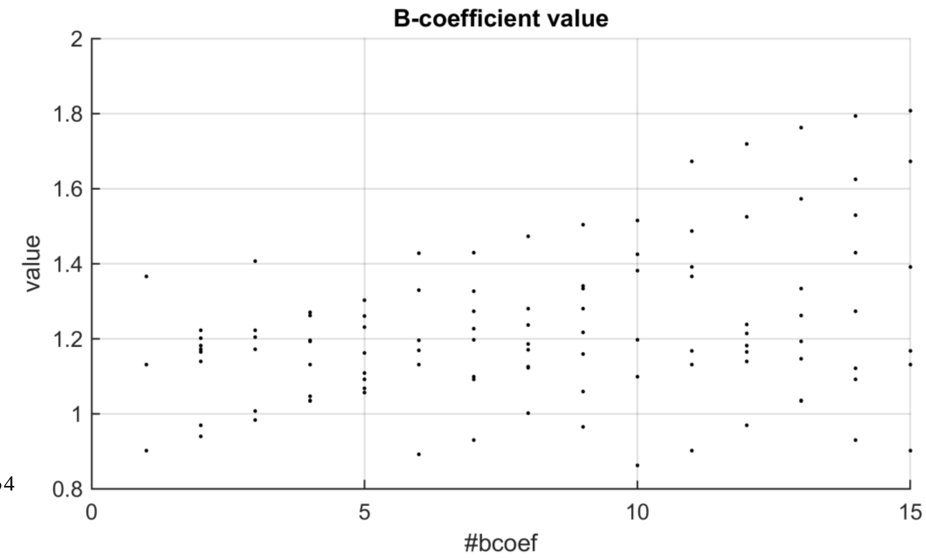
$$p(x) = B^d(b(x)) \cdot \Lambda^{-1} \cdot \hat{c}_P$$

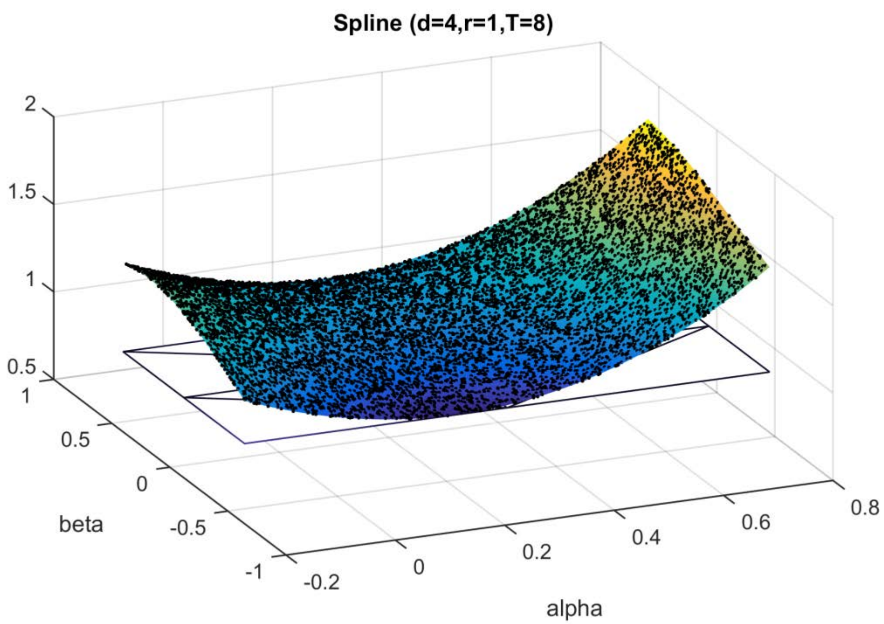
$$c = \begin{bmatrix} \Lambda_{t_1} & 0 & 0 & 0 \\ 0 & \Lambda_{t_2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \Lambda_{t_J} \end{bmatrix}^{-1} \cdot c_p$$

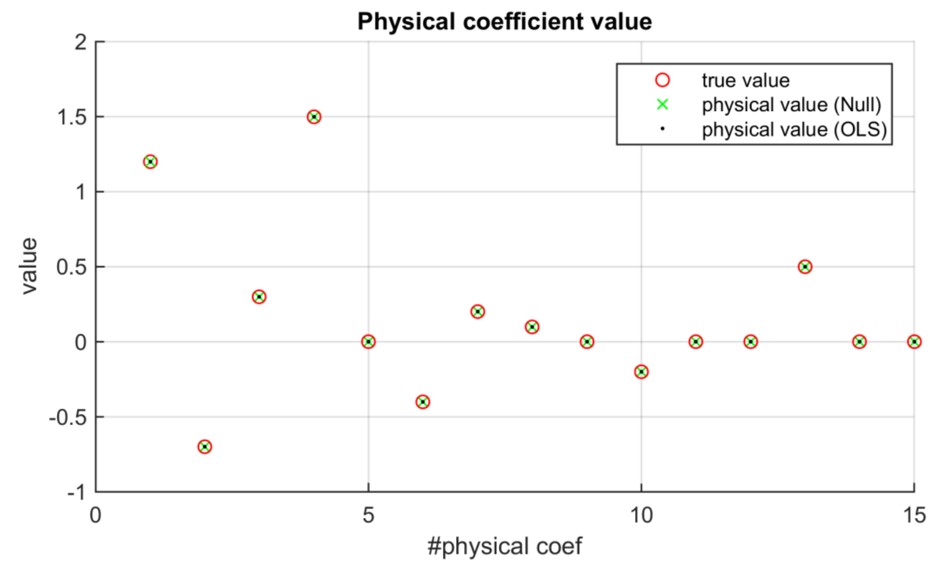
Advanced Applications: Physical Splines

Example: 4^{th} order spline model on 8 triangles:

$$
\begin{aligned}
C_m(\alpha,\beta) = C_{m0} + {} & \\
& C_{m_\alpha}\alpha + C_{m_\beta}\beta + \\
& C_{m_{\alpha^2}}\alpha^2 + C_{m_{\alpha\beta}}\alpha\beta + C_{m_{\beta^2}}\beta^2 + \\
& C_{m_{\alpha^3}}\alpha^3 + C_{m_{\alpha^2\beta}}\alpha^2\beta + C_{m_{\alpha\beta^2}}\alpha\beta^2 + C_{m_{\beta^3}}\beta^3 + \\
& C_{m_{\alpha^4}}\alpha^4 + C_{m_{\alpha^3\beta}}\alpha^3\beta + C_{m_{\alpha^2\beta^2}}\alpha^2\beta^2 + C_{m_{\alpha\beta^3}}\alpha\beta^3 + C_{m_{\beta^4}}\beta^4
\end{aligned}
$$

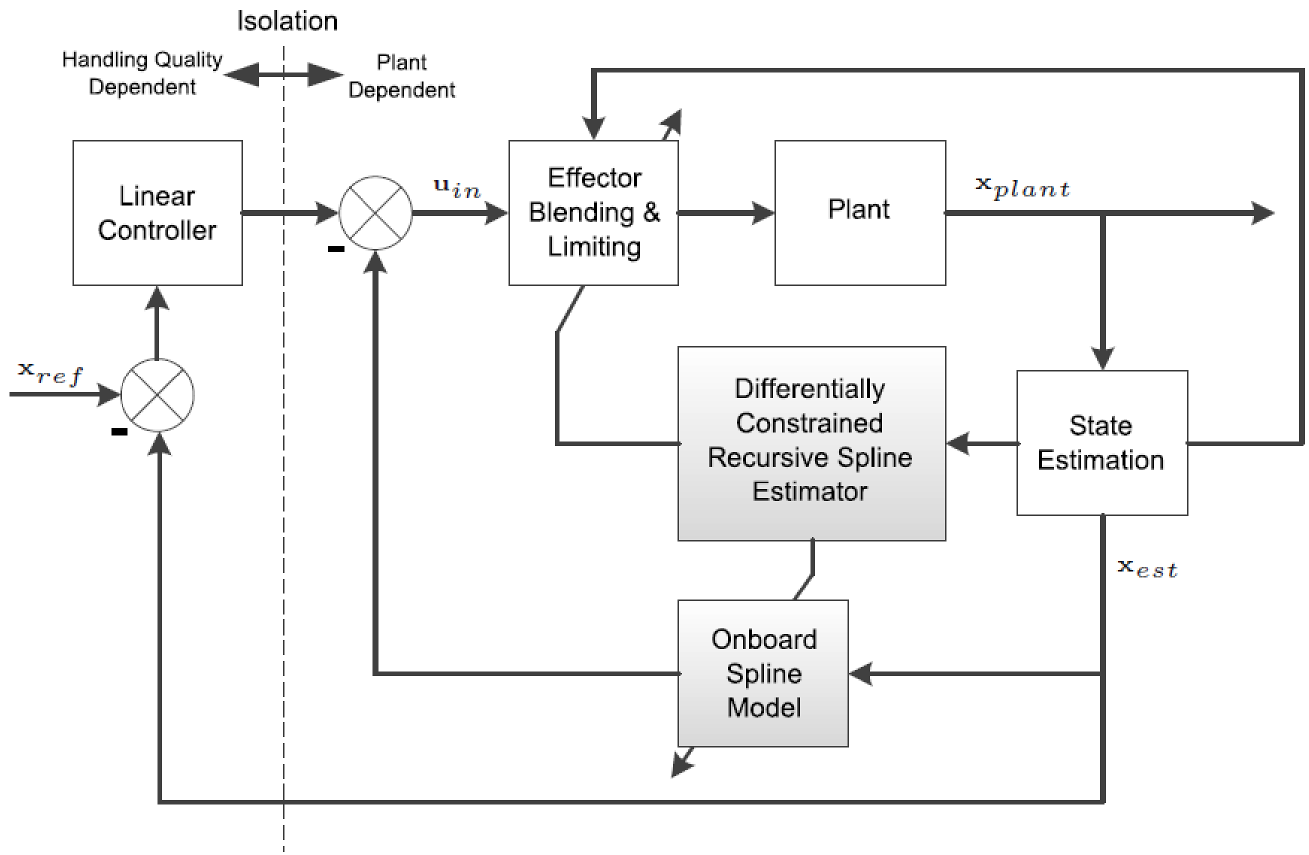






Advanced applications: Adaptive B-splines

Adaptive Simplex Spline based Controller



Advanced applications: Adaptive B-splines

Recursive Equality Constrained Least Squares estimator

Recursive B-coefficient estimator can be used in adaptive model based control systems. It is derived from the OLS (WLS) B-coefficient estimator.

The recursive B-coefficient estimator is:

$$K(t+1) = P(t)x^T(t+1)\left[x(t+1)P(t)x^T(t+1)+\lambda\right]^{-1}$$

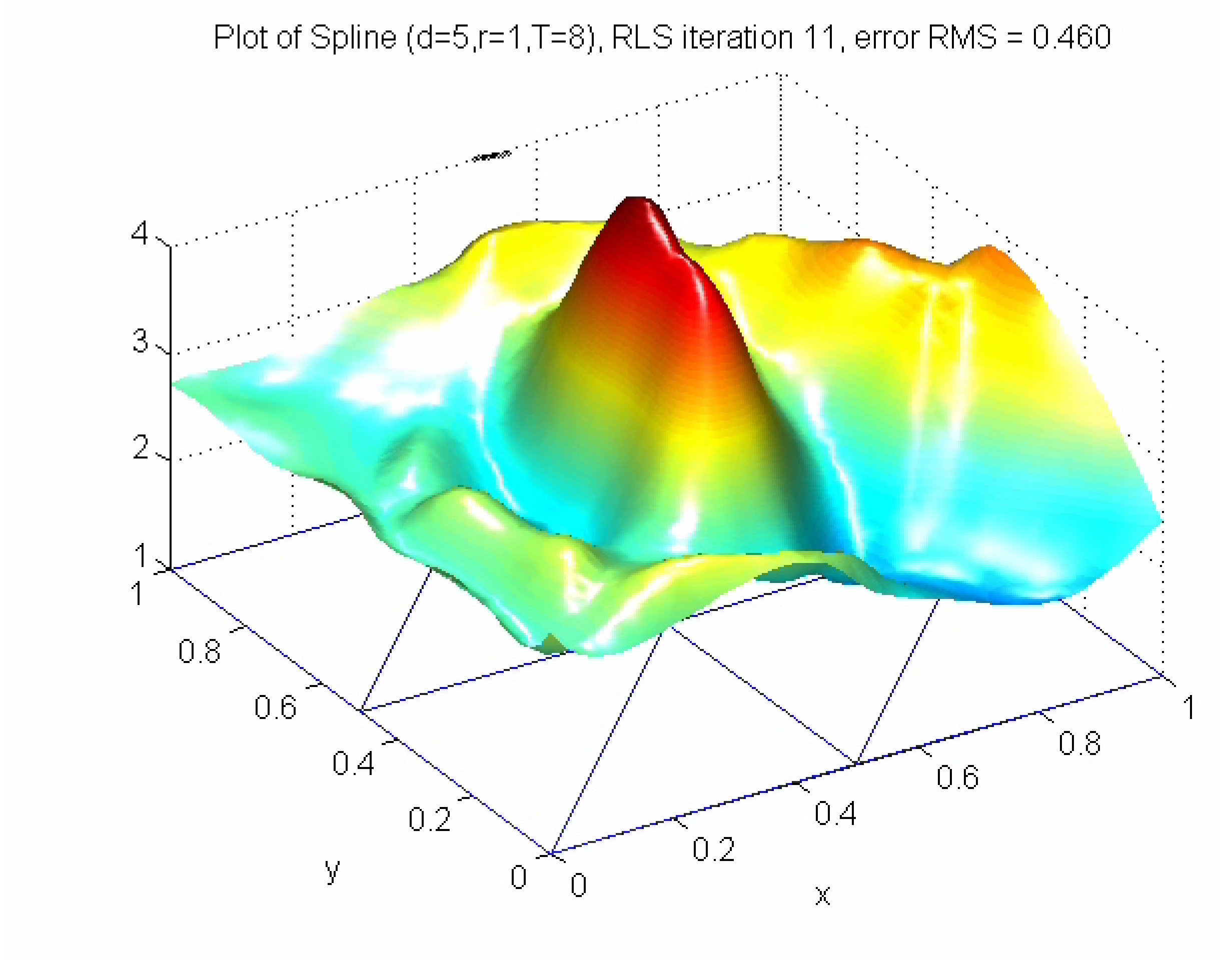
$$P(t+1) = P(t) - K(t+1)x(t+1)P(t)$$

$$\hat{c}(t+1) = \hat{c}(t) + K(t+1)\left[y(t+1) - x(t+1)\hat{c}(t)\right]$$

The smoothness constraints are only used during initialization of the parameter covariance matrix is calculated as follows:

$$P(0) = \left[\left(I - H^+H\right)B^T(0)B(0)\left(I - H^+H\right)\right]^+$$

Advanced applications: Adaptive B-splines



Advanced applications: B-Spline PDE Solvers

Transforming PDE's into differential algebraic equations

We have the vector form of the B-form:

$$p(x) = B^d(b_{t_j}(x)) \cdot c^{t_j}$$

The directional derivatives of B-form polynomials can then be expressed in terms of the **original** B-coefficient vector:

$$D_u^m p(b_{t_j}(x)) = \frac{d!}{(d-m)!} \mathrm{B}^{d-m}(b_{t_j}(x)) \cdot \mathrm{P}^{d,d-m}(a_{t_j}(u)) \cdot \mathrm{c}$$

original B-coefficients

"de Casteljau" matrix

The partial derivatives of the simplex splines are:

$$\frac{\partial^m p(b(x))}{\partial x_i^m} = \frac{d!}{(d-m)!} \mathrm{B}^{d-m}(b_{t_j}(x)) \cdot \mathrm{P}^{d,d-m}(a_{t_j}(x_i)) \cdot \mathrm{c}$$

Advanced applications: B-Spline PDE Solvers

Transforming PDE's into differential algebraic equations

We want to find a solution to the static PDE:

$$\frac{\partial^2 f(x,y)}{dx^2} + \frac{\partial f(x,y)}{dy} = 0$$

Where we assume that the solution of the PDE $f(x,y)$ is approximated with a multivariate B-spline of degree d:

$$f(x,y) \approx B^d(b_{t_j}(x,y)) \cdot c^{t_j}$$

We then have:

$$\frac{\partial^2 f(x,y)}{dx^2} = \frac{d!}{(d-2)!} \mathrm{B}^{d-2}(b_{t_j}(x,y)) \cdot \mathrm{P}^{d,d-2}(a_{t_j}(x)) \cdot \mathrm{c}$$

$$\frac{\partial f(x,y)}{dy} = \frac{d!}{(d-1)!} \mathrm{B}^{d-1}(b_{t_j}(x,y)) \cdot \mathrm{P}^{d,d-1}(a_{t_j}(y)) \cdot \mathrm{c}$$

Advanced applications: B-Spline PDE Solvers

Transforming PDE's into differential algebraic equations

In terms of the simplex splines, we can reformulate

$$\frac{\partial^2 f(x,y)}{dx^2} + \frac{\partial f(x,y)}{dy} = 0$$

into a differential algebraic equation in terms of the B-coefficients:

$$\begin{aligned}
\frac{\partial^2 f(x,y)}{dx^2} + \frac{\partial f(x,y)}{dy} &= \frac{d!}{(d-2)!}\mathrm{B}^{d-2}(b_{t_j}(x,y)) \cdot \mathrm{P}^{d,d-2}(a_{t_j}(x)) \cdot \mathrm{c} + \\
&\quad + \frac{d!}{(d-1)!}\mathrm{B}^{d-1}(b_{t_j}(x,y)) \cdot \mathrm{P}^{d,d-1}(a_{t_j}(y)) \cdot \mathrm{c} \\
&= \left(\frac{d!}{(d-2)!}\mathrm{B}^{d-2}(b_{t_j}(x,y)) \cdot \mathrm{P}^{d,d-2}(a_{t_j}(x)) + \frac{d!}{(d-1)!}\mathrm{B}^{d-1}(b_{t_j}(x,y)) \cdot \mathrm{P}^{d,d-1}(a_{t_j}(y)) \right) \cdot \mathrm{c} \\
&= \left(\Omega^{d,2}(x,y) + \Omega^{d,1}(x,y) \right) \cdot \mathrm{c}
\end{aligned}$$

Advanced applications: B-Spline PDE Solvers

Transforming PDE's into differential algebraic equations

Let's assume we have **measurements** made on the partial derivatives:

$$\left.\frac{\partial^2 f(x_i, y_i)}{dx^2}\right|_{x_i, y_i} \rightarrow F_2,$$

$$\left.\frac{\partial f(x_i, y_i)}{dy}\right|_{x_i, y_i} \rightarrow F_1$$

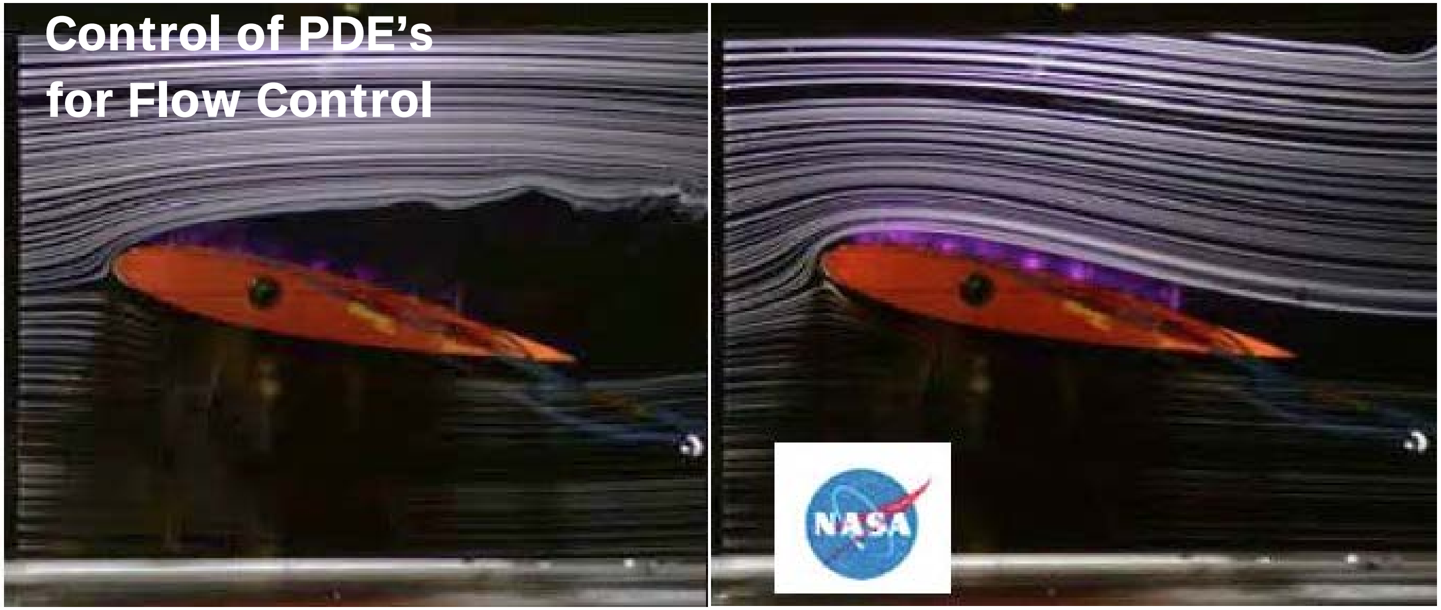
Then we can simply estimate the B-coefficients of the spline function that approximates the PDE solution as follows:

$$\hat{c}^{t_j} = \left[\left(\Omega^{d,2}(x,y)+\Omega^{d,1}(x,y)\right)^T \left(\Omega^{d,2}(x,y)+\Omega^{d,1}(x,y)\right)\right]^{-1} \left(\Omega^{d,2}(x,y)+\Omega^{d,1}(x,y)\right)^T \left(F_2 + F_1\right)$$

The approximated solution to the PDE then is:

$$f(x,y) \approx B^d(b_{t_j}(x,y)) \cdot \hat{c}^{t_j}$$

Current Spline Research



Given the linear PDE: $$\frac{\partial y(x,t)}{dt} = a_0 + \sum_{i=1}^{n} a_i \frac{\partial^i y(x,t)}{dx^i} + g(x)u(t)$$

Galerkin projection based spline discretization:

$$y(x,t) = B^d(x)c(t) \rightarrow \frac{\partial y(x,t)}{dt} = B^d(x)\dot{c}(t);\ a_i \frac{\partial^i y(x,t)}{dx^i} = \frac{d!}{(d-n)!} B^{d-i} P_{x^i}^{d,d-i} c(t)$$

...allows solution to be written as a state space system: $\dot{c}(t) = A \cdot c(t) + F \cdot u(t)$

H. Tol, C.C. de Visser, Model Reduction and Control of Linear Partial Differential Equations using Multivariate Splines, 2016.

Current Spline Research

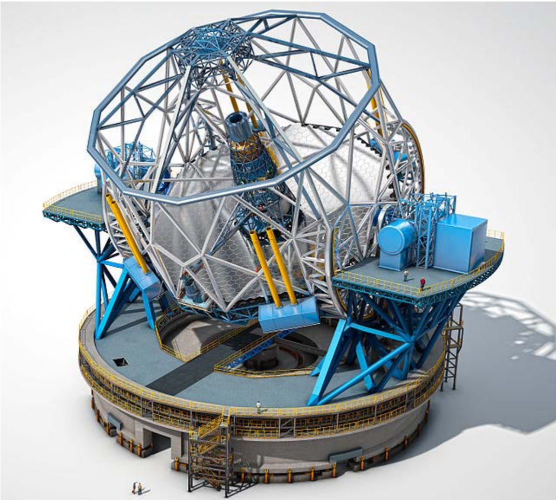
Multivariate Splines are Hot!

- H.J. Tol, M. Kotsonis, C.C. de Visser, Experimental model-based estimation and control of natural Tollmien-Schlichting waves, *AIAA Journal*, **2019**
- H.J. Tol, M. Kotsonis, C.C. de Visser, Pressure based estimation and control of TS-waves in Falkner-Skan boundary layers, *AIAA Journal*, **2018**
- E. Brunner, C.C. de Visser, C. Vuik, M. Verhaegen, GPU implementation for spline-based wavefront reconstruction, *JOSA-A*, **2018**.
- H.J. Tol, M. Kotsonis, C.C. de Visser, B. Bamieh, Localized estimation and control of linear instabilities in 2-D wall-bounded shear flows, *Journal of Fluid Mechanics*, **2017**
- E. Brunner, C.C. de Visser, M. Verhaegen. Nonlinear Spline Wavefront Reconstruction from Shack-Hartmann Intensity Measurements through Small Aberration Approximations. *JOSA-A*, **2017**.
- L.G. Sun, C.C. de Visser, and Q.P. Chu, Improving Flexibility of Multivariate Spline Model Structures for Aerodynamic Modelling, *Journal of Aerospace Engineering*, **2017**.
- M. Viegers, E. Brunner, C.C. de Visser, M. Verhaegen. Nonlinear Spline Wavefront Reconstruction through Moment-based Shack-Hartmann Sensor Measurements, *Optics Express*, **2017**
- H. Tol, C.C. de Visser, Model Reduction and Control of Linear Partial Differential Equations using Multivariate Splines, *International Journal of Control*, **2016**.
- H. Tol, C.C. de Visser, E. van Kampen and Q.P. Chu, Adaptive Multivariate Spline Based Control Allocation for High Performance Aircraft, *AIAA Journal of Guidance, Control, and Dynamics*, **2016**.
- C.C. de Visser, E. Brunner, and M. Verhaegen, On Distributed Wavefront Reconstruction for Large Scale Adaptive Optics Systems, *JOSA-A*, **2016**.
- N. Govindarajan, C. C. de Visser and K. Krishnakumar, A sparse collocation method for solving time-dependent HJB equations using multivariate B-splines, *Automatica*, **2014**.
- H. Tol, C.C. de Visser, E. van Kampen and Q.P. Chu, Nonlinear Multivariate Spline Based Control Allocation for High Performance Aircraft, *AIAA Journal of Guidance, Control, and Dynamics*, **2014**.
- L.G. Sun, C.C. de Visser, Q.P. Chu, and J.A. Mulder, Online Aerodynamic Model Identification using a Recursive Sequential Method for Multivariate Splines, *AIAA Journal of Guidance, Control, and Dynamics*, **2013**.
- C.C. de Visser and M. Verhaegen, Wavefront reconstruction in adaptive optics systems using nonlinear multivariate splines, *JOSA-A*, **2013**.
- C.C. de Visser, E. van Kampen, Q.P. Chu, and J.A. Mulder, Intersplines: A New Approach to Globally Optimal Multivariate Splines Using Interval Analysis, *Reliable Computing*, **2012**.
- C.C. de Visser, Q.P. Chu, and J.A. Mulder, Differential constraints for bounded recursive identification with multivariate splines. *Automatica*, **2011**.
- C.C. de Visser, Q.P. Chu, and J.A. Mulder, A new approach to linear regression with multivariate splines. *Automatica*, **2009**.

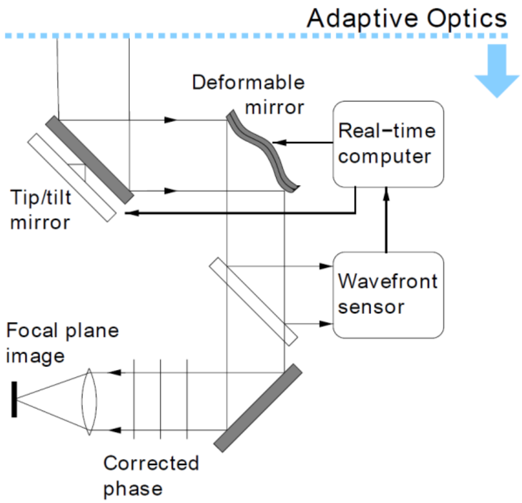
Current Spline Research

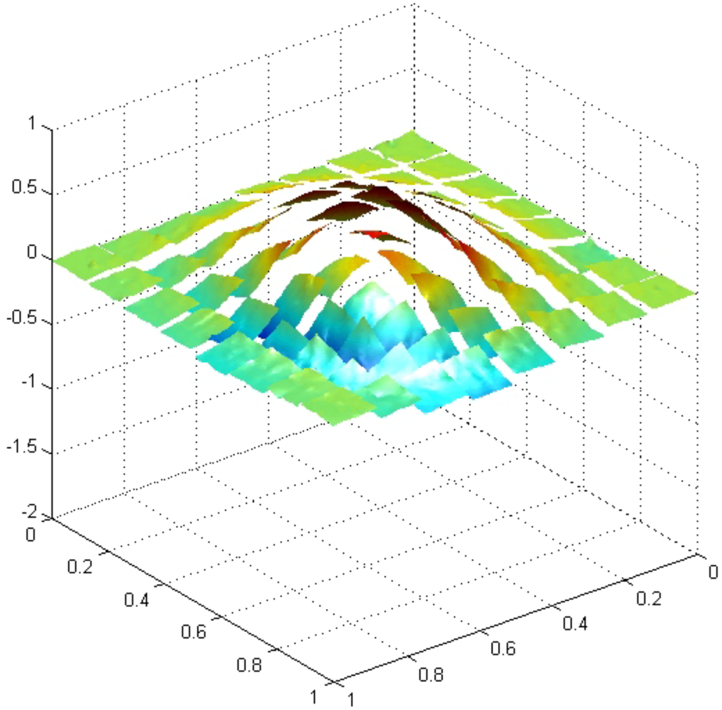
SABRE / D-SABRE

Multivarate B-spline based distributed wavefront reconstruction for adaptive optics in Extremely Large Telescopes



ELT: 39m primary mirror, 40000 actuators, 3000Hz update rate



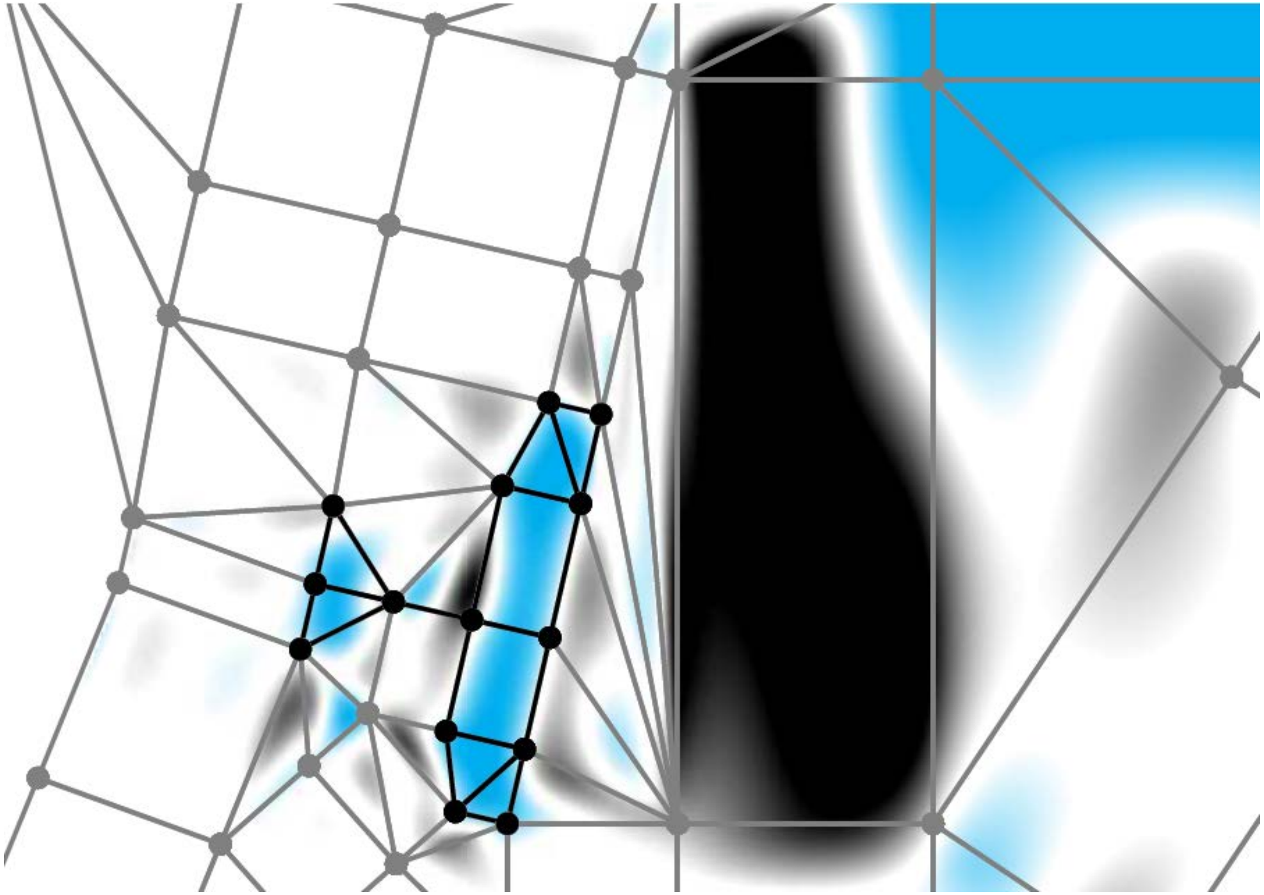


- C.C. de Visser, E. Brunner, and M. Verhaegen, On Distributed Wavefront Reconstruction for Large Scale Adaptive Optics Systems, *JOSA-A*, **2016**.

Current Spline Research

Visser's work

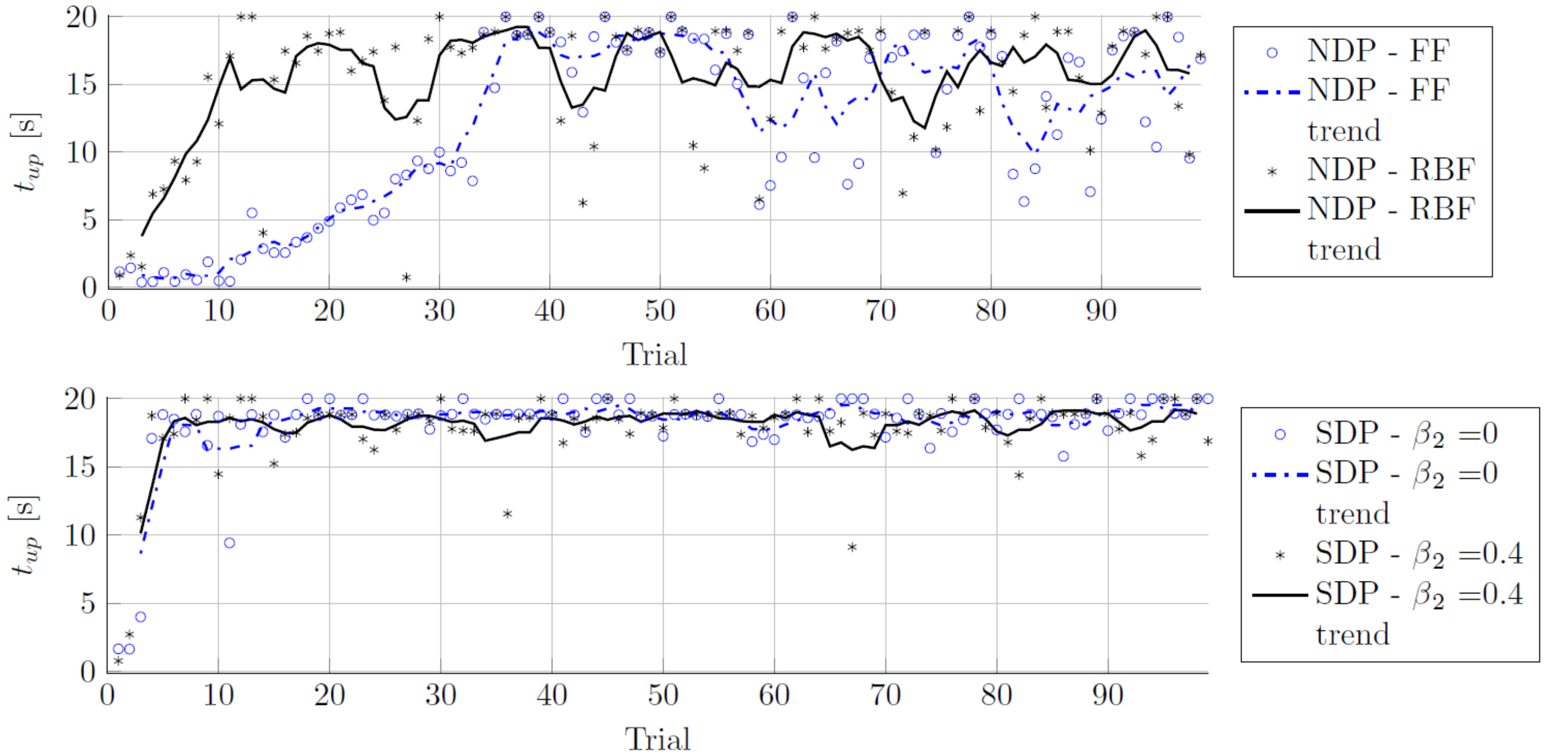
Generalization of simplex splines to general geometries: **Simplotope Splines**



Current Spline Research

Eerland's work

Machine learning using multivariate simplex B-splines

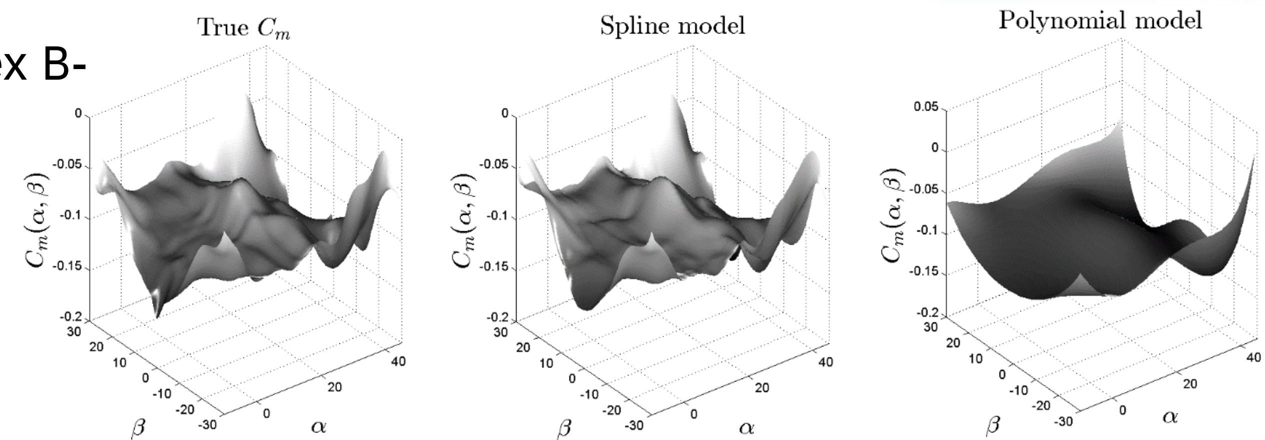


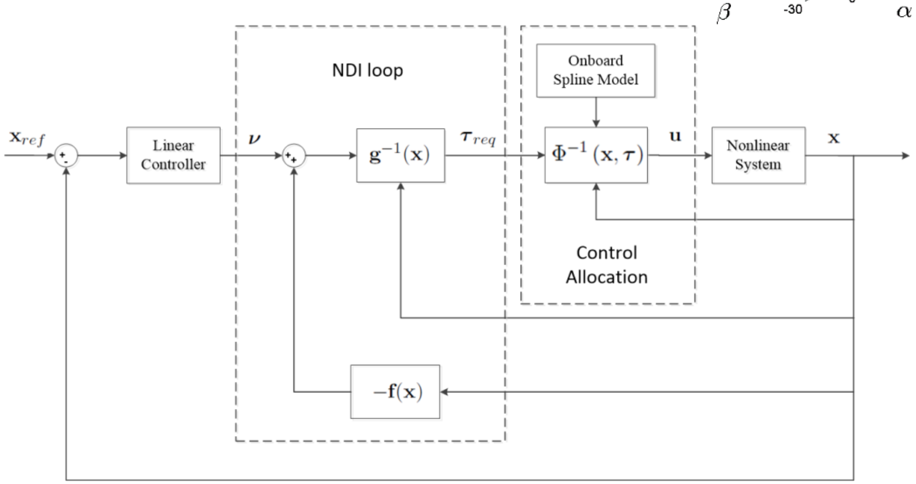
W. Eerland, C. C. de Visser, E. van Kampen, On Approximate Dynamic Programming with Multivariate Splines for Adaptive Control, Neurocomputing, **in preparation**, 2016.

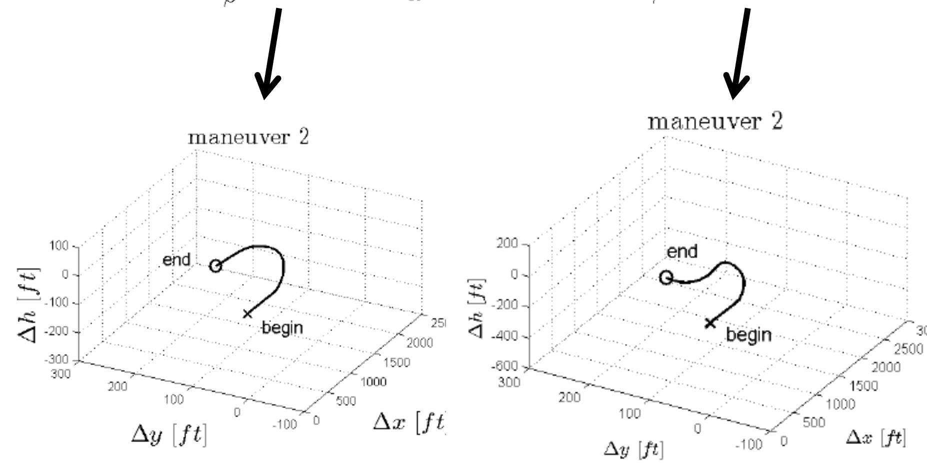
Current Spline Research

Henry Tol's work

Nonlinear adaptive control using simplex B-spline based control allocation.







H. Tol, C.C. de Visser, E. van Kampen and Q.P. Chu, Nonlinear Multivariate Spline Based Control Allocation for High Performance Aircraft, *AIAA Journal of Guidance, Control, and Dynamics*, 2015.

Current Spline Research

Nonlinear control allocation for over-actuated aircraft

INCA control law:

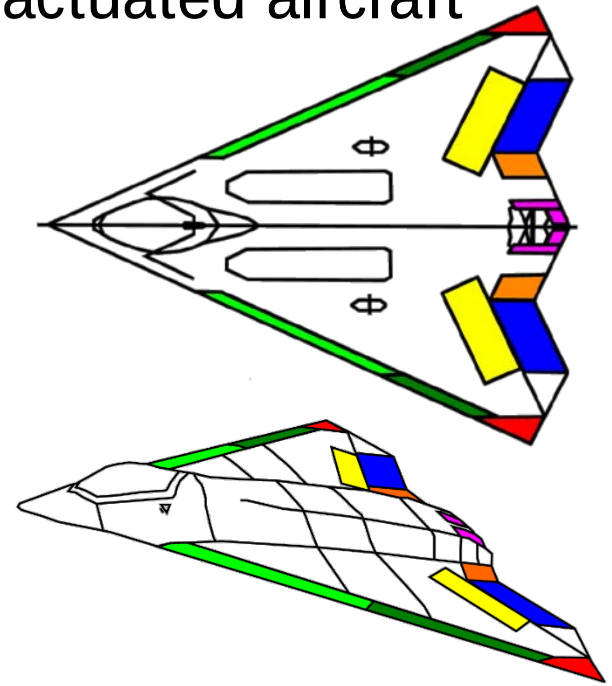
$$\Delta\delta_c = \left[\frac{\partial\phi(x_0,\delta_0)}{\partial\delta}\right]^{+} g(x)^{-1}(v(x)-\dot{x}_0)$$

effector increment vector

control effectiveness spline function

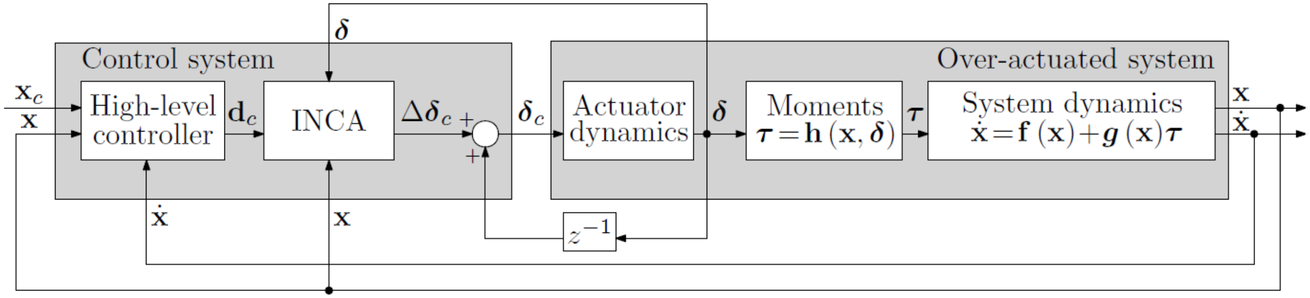
inverse inertia matrix

command error



Lockheed-Martin ICE model (control effectors highlighted)

Controller architecture:



Future Spline Research

- Efficient flight envelope modelling
  - Based on flight envelope parametrization using simplex B-splines
- Simplotope B-splines for aero-model identification
  - Allows use of a-priori information for model structure selection
  - Allows efficient **distributed** identification methods
- Control of PDE's
  - Uses Simplex Splines to discretize infinite dimensional systems into simple to use state space descriptions.
  - Many interesting applications in flow control, wind energy, etc...
- Adaptive nonlinear control
  - Uses simplex B-splines with smart local forgetting/learning to modify aerodynamic model after failures.

Goals of this Lecture

Questions that were answered during this lecture:

1. *What is the relation between a spline and a polynomial?*
   - *A spline is a piecewise defined polynomial function with a predefined continuity between the polynomial pieces.*
2. *What are the different types of multivariate splines?*
   - *There are many types of multivariate splines. The best known multivariate spline types are multivariate tensor product splines and thin-plate splines.*
3. *What are multivariate simplex B-splines?*
   - *Multivariate simplex B-splines are a general type of multivariate spline with basis functions defined on simplices in terms of barycentric coordinates.*

Goals of this Lecture

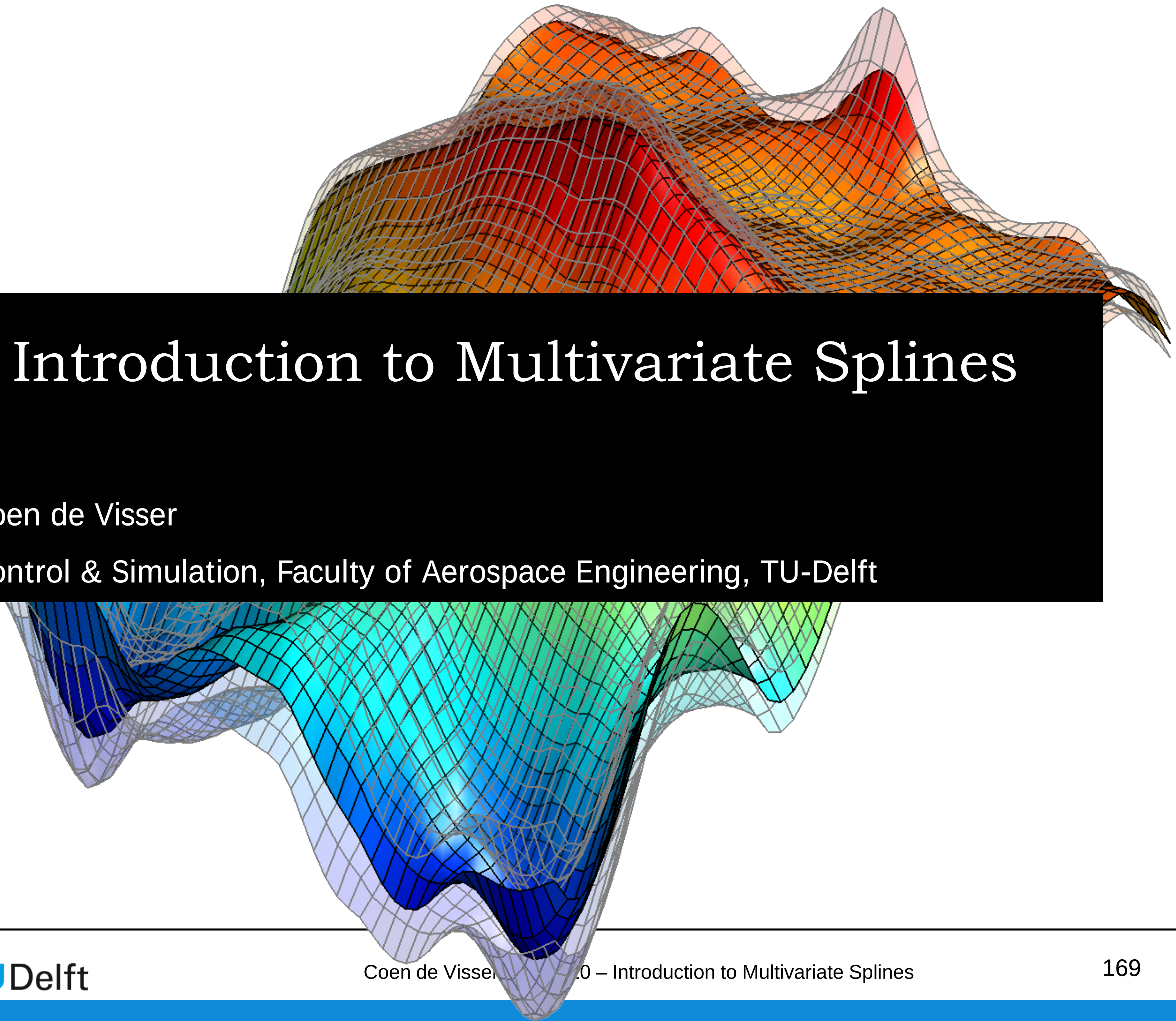
Questions that were answered during this lecture:

4. *What are the unique features and advantages of multivariate simplex B-splines?*
   - *They are defined for any number of dimensions.*
   - *They are linear in the parameters which allows the use of efficient linear solvers.*
   - *They have stable local basis functions.*
   - *Their basis functions are defined in terms of barycentric coordinates on simplices.*
   - *Their B-coefficients have a spatial location within the simplex which allows local model modification and local model quality assessment.*
   - *Local approximation power can be increased by locally increasing triangulation density.*

Goals of this Lecture

Questions that were answered during this lecture:

5. *How do we use simplex B-splines in aerospace applications?*
   - *They are compatible with standard linear regression approaches for system identification.*
   - *Both offline and online (recursive) system identification approaches can be used to estimate the B-coefficients.*
   - *They can be used as internal model in a model-based control system.*
   - *They can be used as adaptive element in an adaptive controller.*
   - *They can be used to discretize PDEs allowing for PDE solving and control.*



Introduction to Multivariate Splines

Coen de Visser

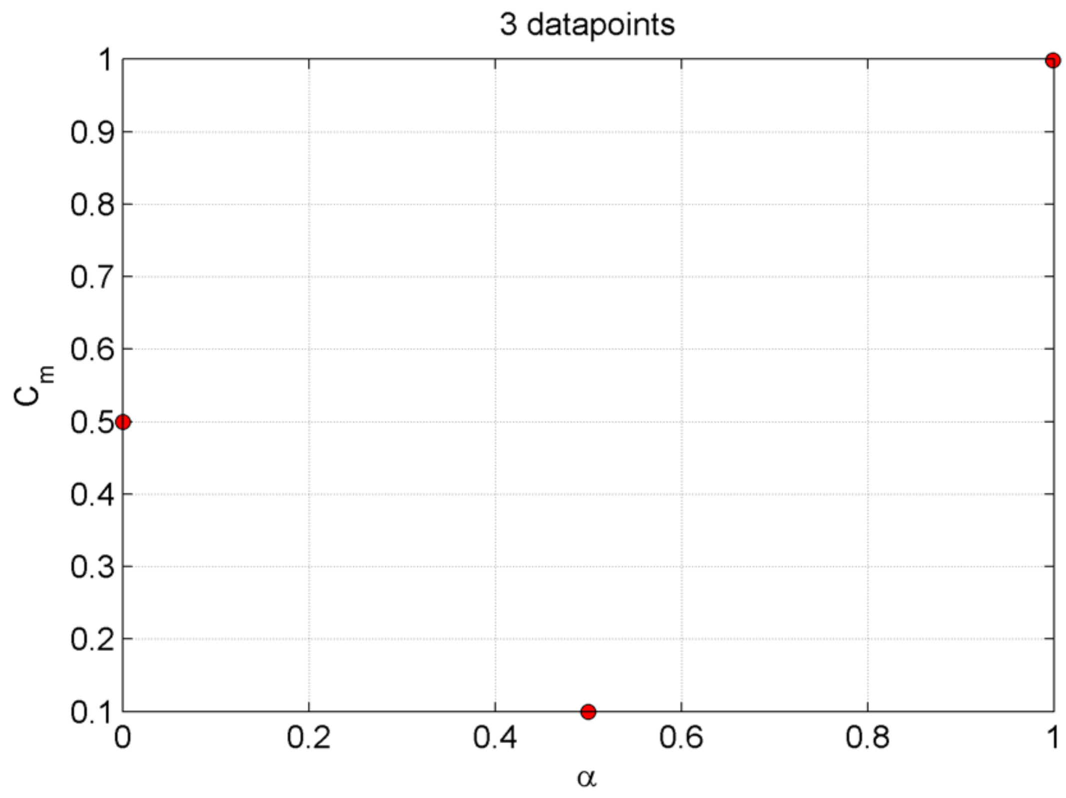
Control & Simulation, Faculty of Aerospace Engineering, TU-Delft

Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

$$C_m = \begin{bmatrix} 0.5 \\ 0.1 \\ 1 \end{bmatrix},$$

$$\alpha = \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix}$$

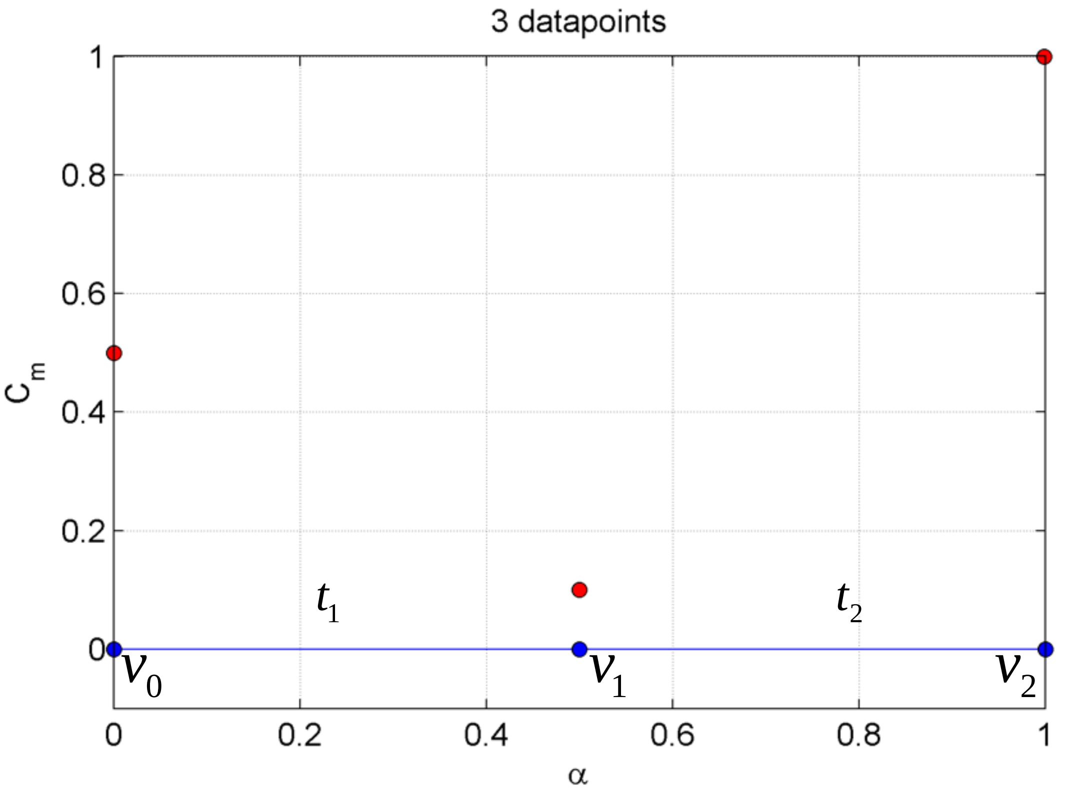


Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

Step 1: Define triangulation

$$t_1 = \langle v_0, v_1 \rangle = \langle 0, 0.5 \rangle$$
$$t_2 = \langle v_1, v_2 \rangle = \langle 0.5, 1.0 \rangle$$



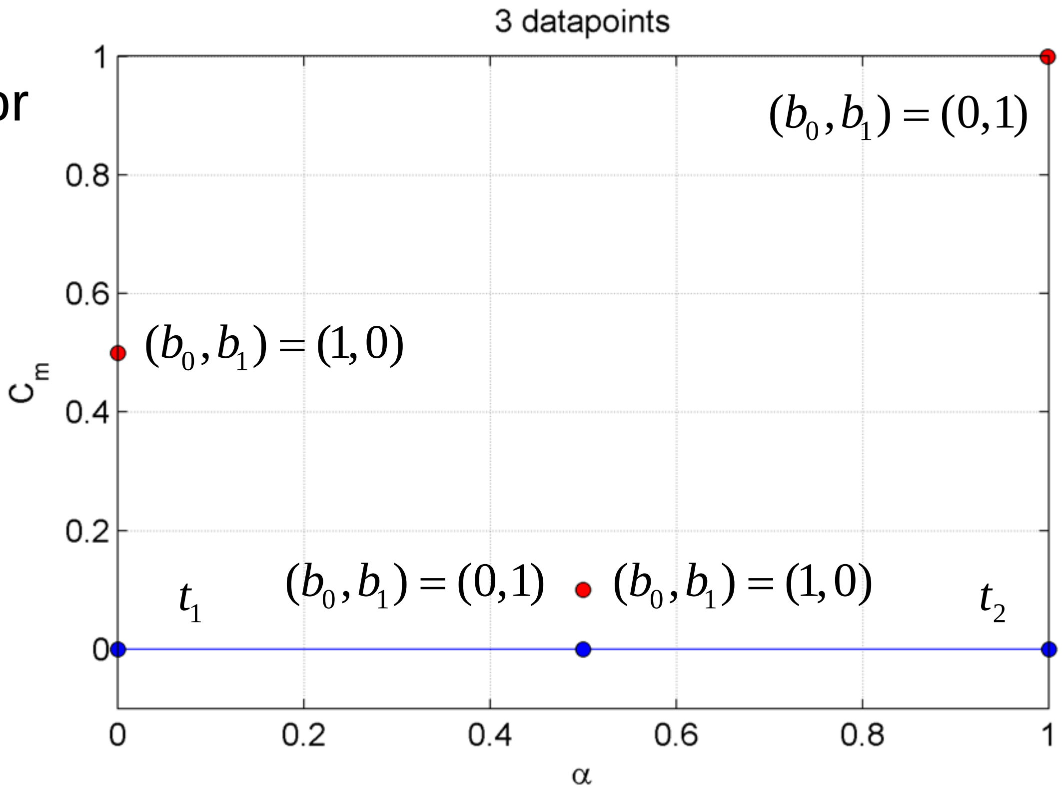
Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

Step 2: Calculate barycentric coordinates of data points for each simplex:

$$t_1 : \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} (1,0) \\ (0,1) \end{bmatrix}$$

$$t_2 : \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} (1,0) \\ (0,1) \end{bmatrix}$$



Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

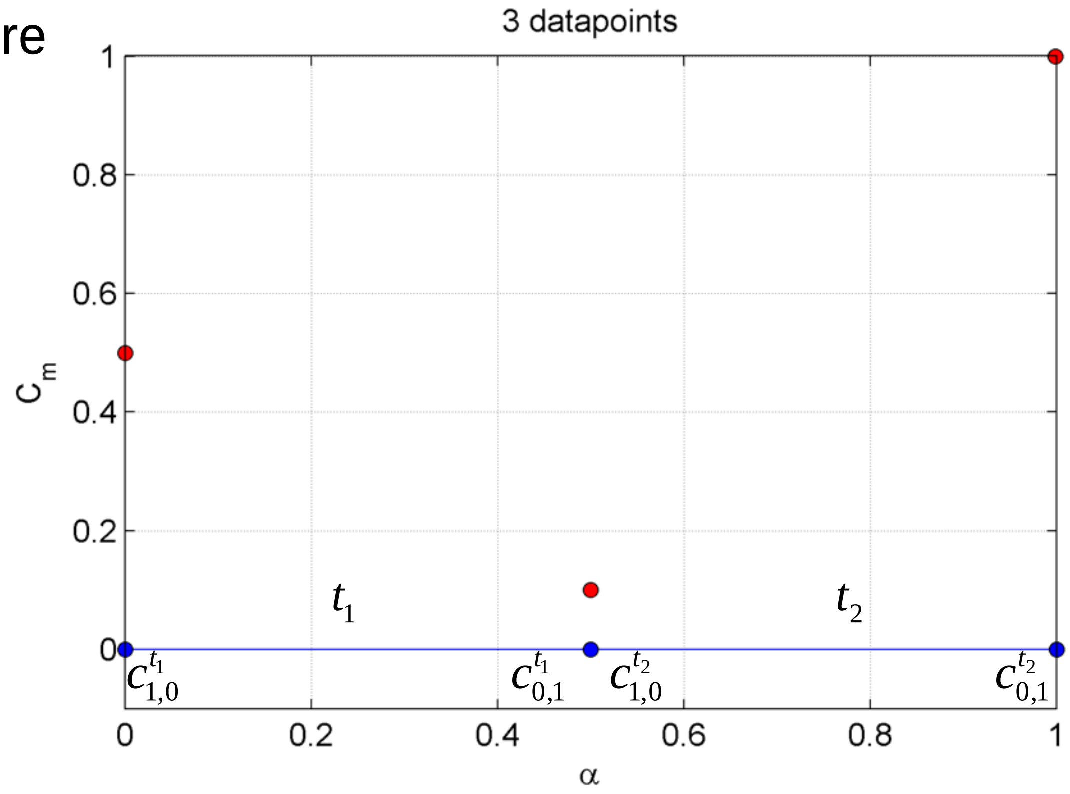
Step 3: Define model structure

$$Y^{t_1} = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix}, \qquad Y^{t_2} = \begin{bmatrix} 0.5 \\ 1.0 \end{bmatrix}$$

$$X^{t_1} = \begin{bmatrix} b_0(1) & b_1(1) \\ b_0(2) & b_1(2) \end{bmatrix},$$

$$X^{t_2} = \begin{bmatrix} b_0(1) & b_1(1) \\ b_0(2) & b_1(2) \end{bmatrix},$$

$$c^{t_1} = \begin{bmatrix} c^{t_1}_{1,0} \\ c^{t_1}_{0,1} \end{bmatrix}, \qquad c^{t_2} = \begin{bmatrix} c^{t_2}_{1,0} \\ c^{t_2}_{0,1} \end{bmatrix}$$

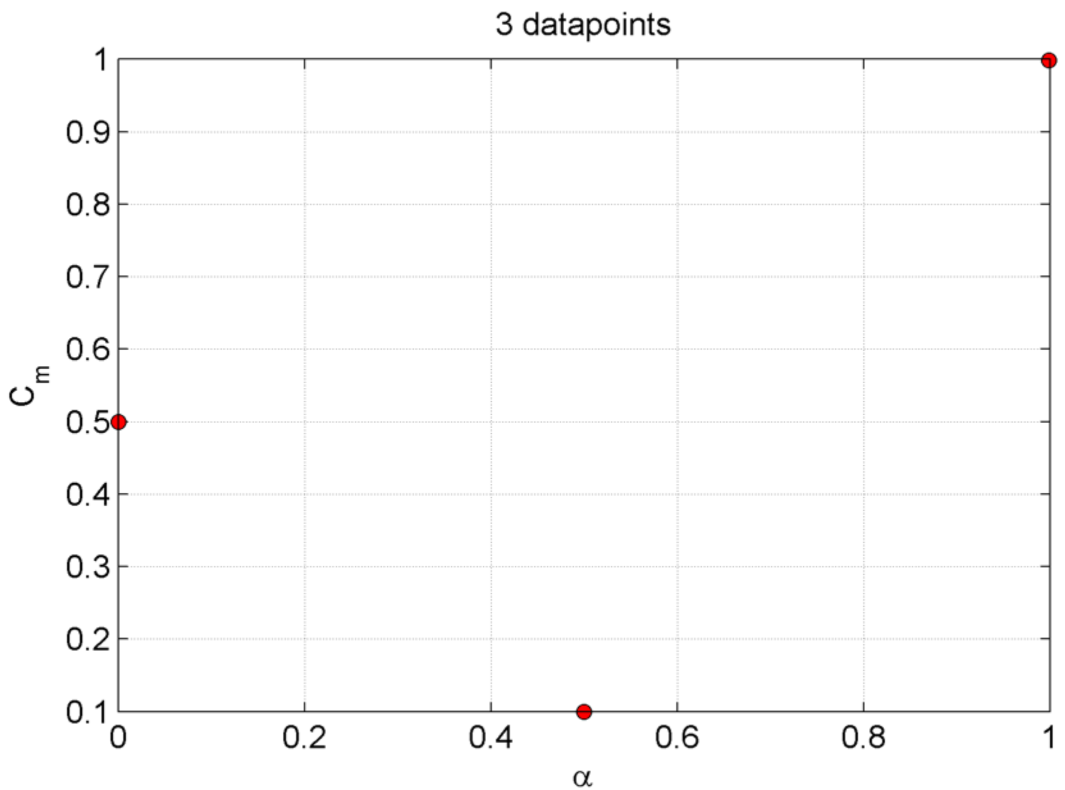


Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

$$C_m = \begin{bmatrix} 0.5 \\ 0.1 \\ 1 \end{bmatrix},$$

$$\alpha = \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix}$$

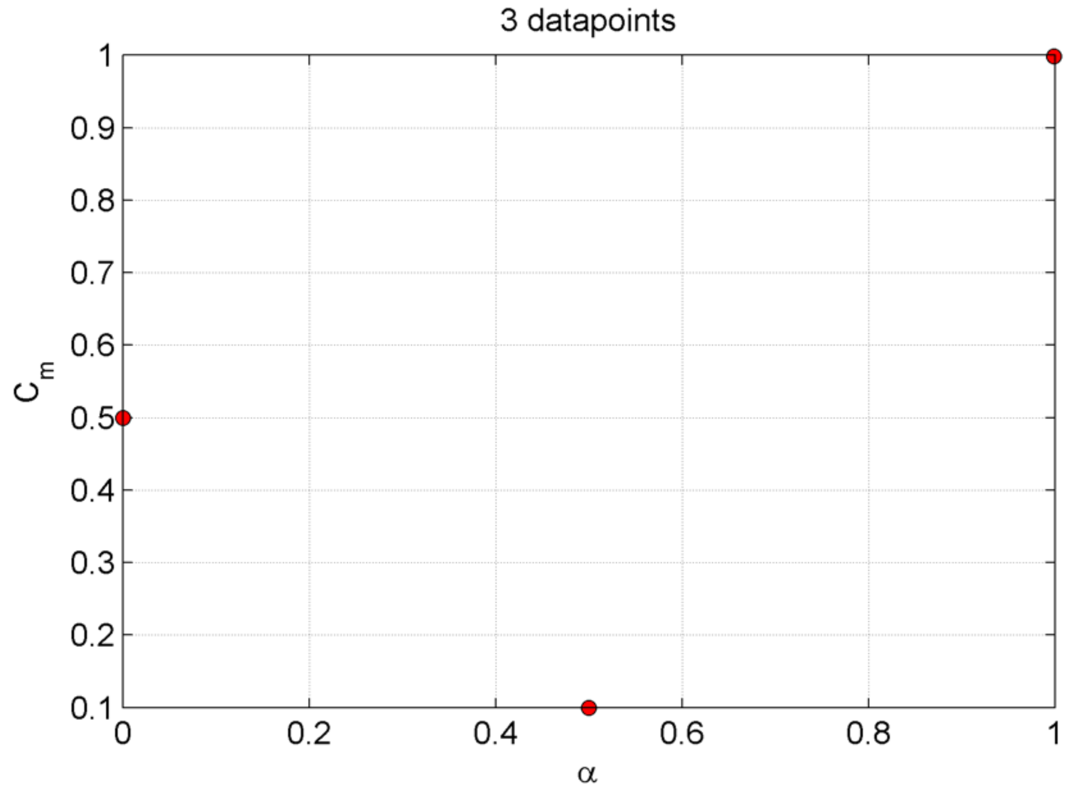


Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

$$C_m = \begin{bmatrix} 0.5 \\ 0.1 \\ 1 \end{bmatrix},$$

$$\alpha = \begin{bmatrix} 0 \\ 0.5 \\ 1 \end{bmatrix}$$

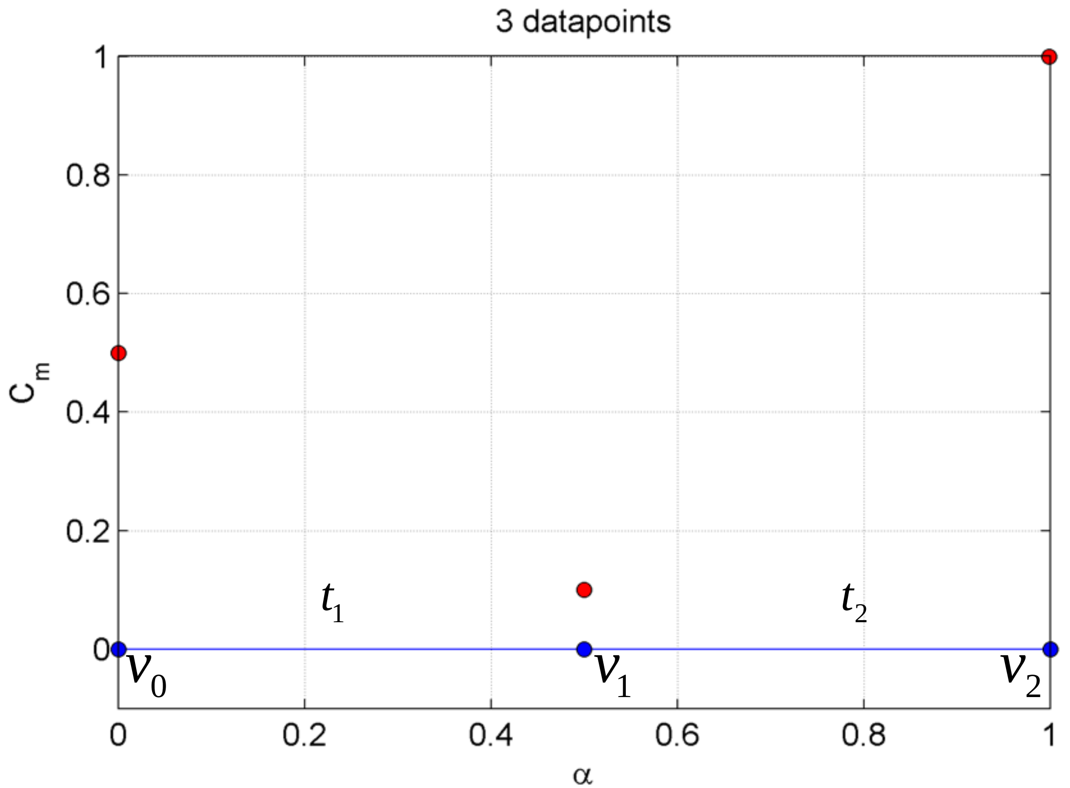


Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

Step 1: Define triangulation

$$t_1 = \langle v_0, v_1 \rangle = \langle 0, 0.5 \rangle$$
$$t_2 = \langle v_1, v_2 \rangle = \langle 0.5, 1.0 \rangle$$



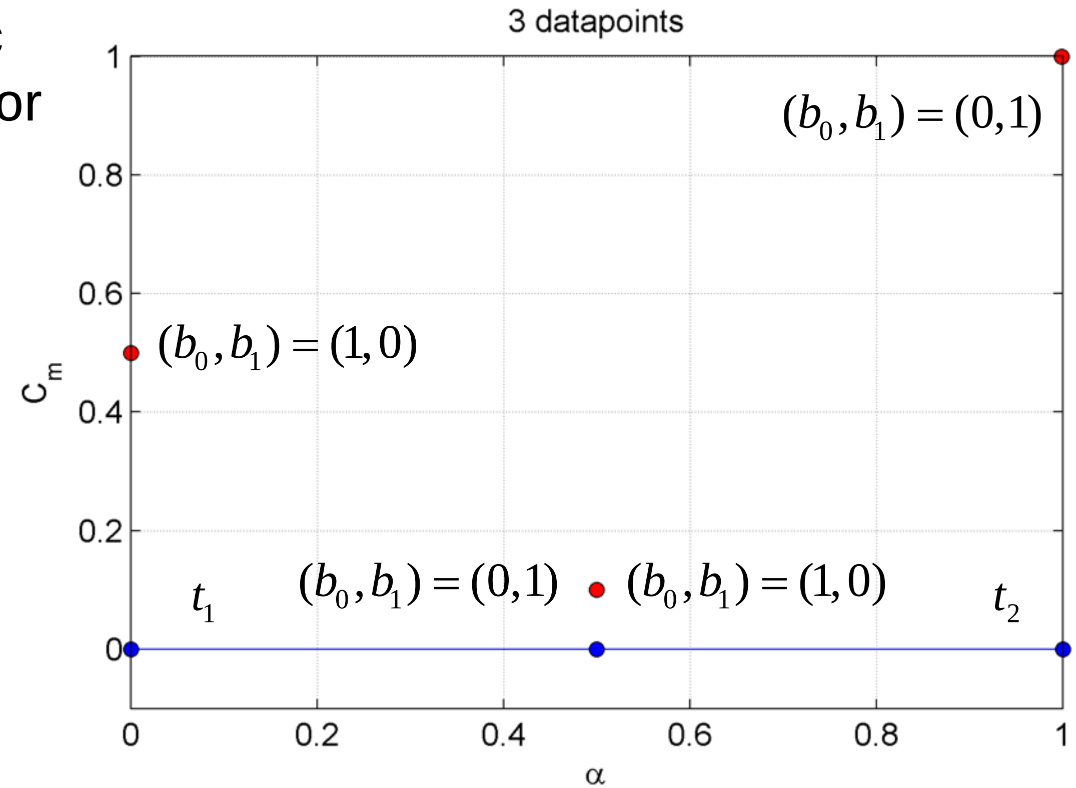
Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

Step 2: Calculate barycentric coordinates of data points for each simplex:

$$t_1 : \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} (1,0) \\ (0,1) \end{bmatrix}$$

$$t_2 : \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} (1,0) \\ (0,1) \end{bmatrix}$$



Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

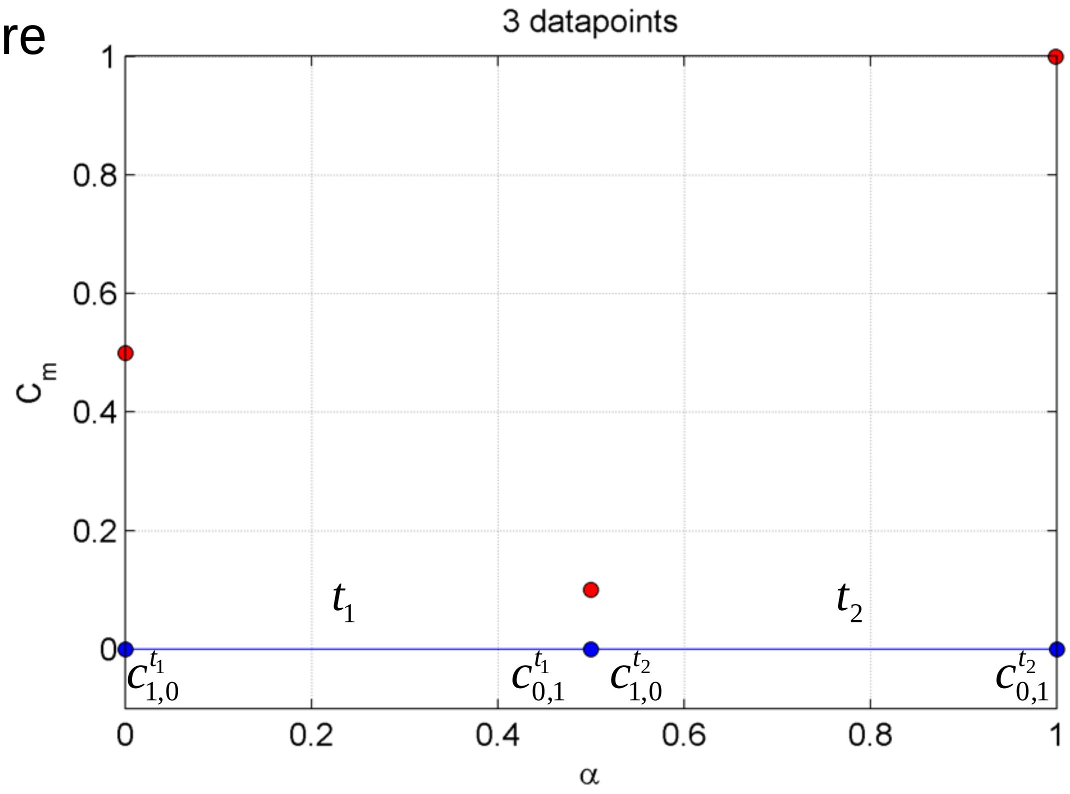
Step 3: Define model structure

$$Y^{t_1} = \begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix}, \qquad Y^{t_2} = \begin{bmatrix} 0.5 \\ 1.0 \end{bmatrix}$$

$$X^{t_1} = \begin{bmatrix} b_0(1) & b_1(1) \\ b_0(2) & b_1(2) \end{bmatrix},$$

$$X^{t_2} = \begin{bmatrix} b_0(1) & b_1(1) \\ b_0(2) & b_1(2) \end{bmatrix},$$

$$c^{t_1} = \begin{bmatrix} c^{t_1}_{1,0} \\ c^{t_1}_{0,1} \end{bmatrix}, \qquad c^{t_2} = \begin{bmatrix} c^{t_2}_{1,0} \\ c^{t_2}_{0,1} \end{bmatrix}$$



Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

Step 4: Define smoothness matrix

$$c_{0,1}^{t_1} = \sum_{|\gamma|=0} c_{(\kappa_0,0)+\gamma}^{t_2} B_{\gamma}^{0}(v_0),$$

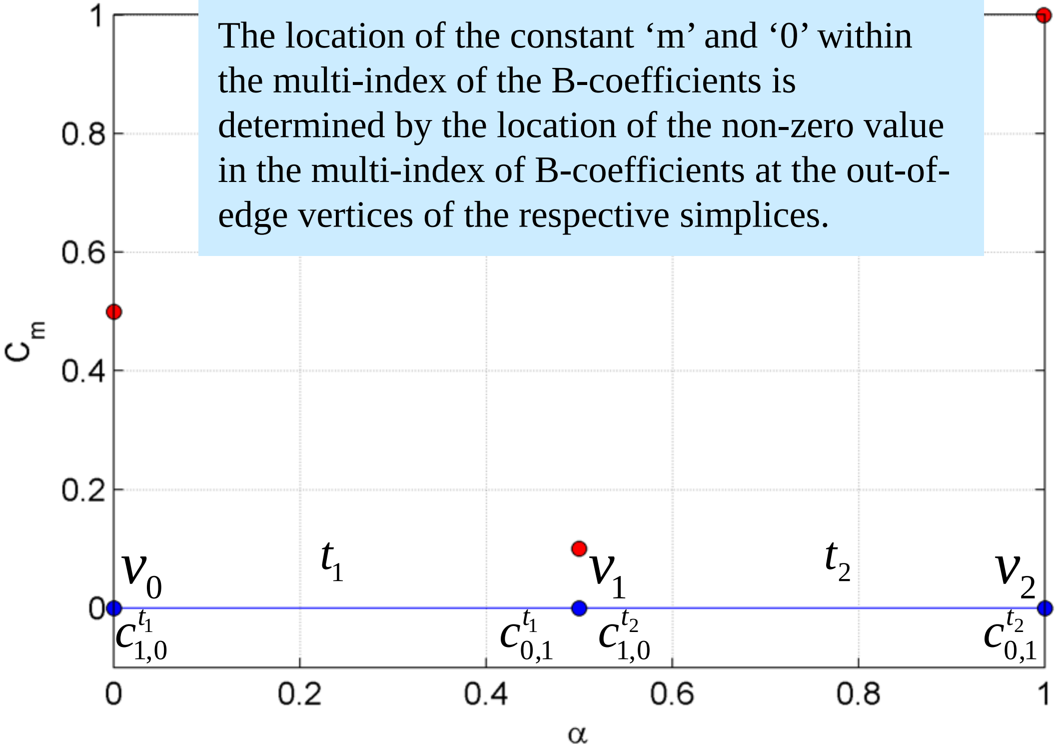
$$c_{0,1}^{t_1} = c_{1,0}^{t_2}$$

$$\Rightarrow 0 = \begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_{1,0}^{t_1} \\ c_{0,1}^{t_1} \\ c_{1,0}^{t_2} \\ c_{0,1}^{t_2} \end{bmatrix}$$

$$= H \cdot c$$

Generalizing the equations of continuity

The location of the constant ‘m’ and ‘0’ within the multi-index of the B-coefficients is determined by the location of the non-zero value in the multi-index of B-coefficients at the out-of-edge vertices of the respective simplices.



Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

Step 5: Solve constrained linear regression problem:

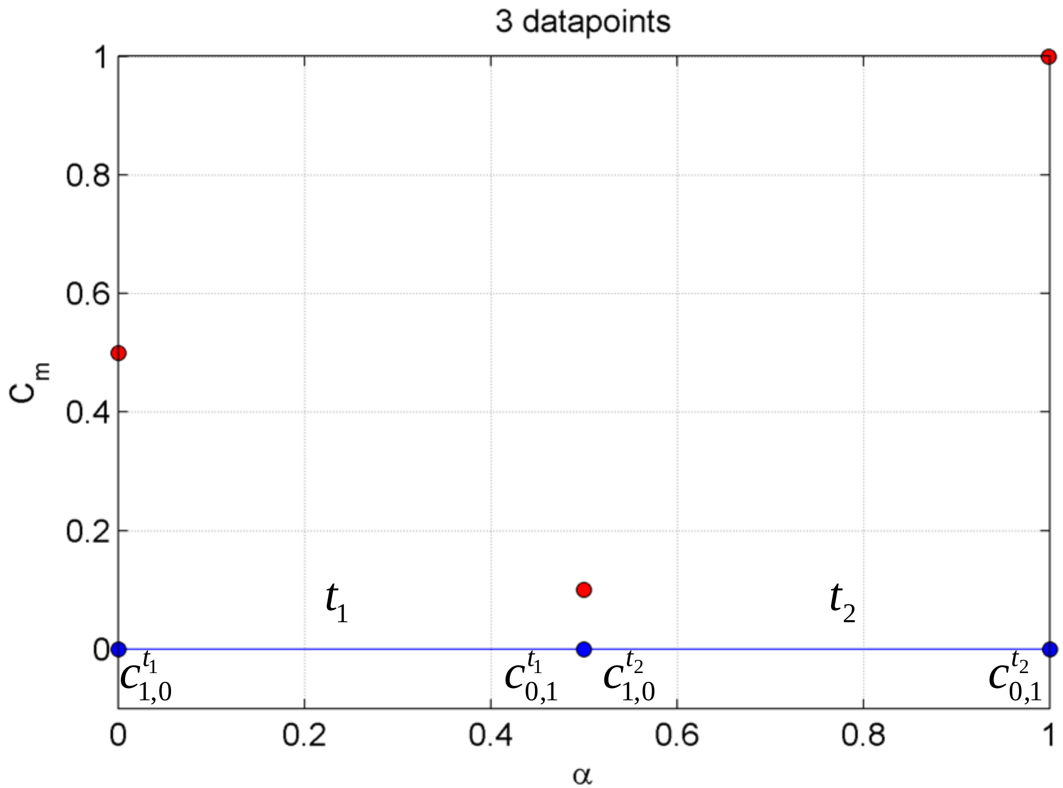
$$Y = \begin{bmatrix} Y^{t_1} \\ Y^{t_2} \end{bmatrix}, X = \begin{bmatrix} X^{t_1} & 0 \\ 0 & X^{t_2} \end{bmatrix}$$

$$c = \begin{bmatrix} c^{t_1} \\ c^{t_2} \end{bmatrix}$$

$$Y = X \cdot c + \varepsilon$$

Subject to

$$0 = H \cdot c$$



Linear regression with simplex splines

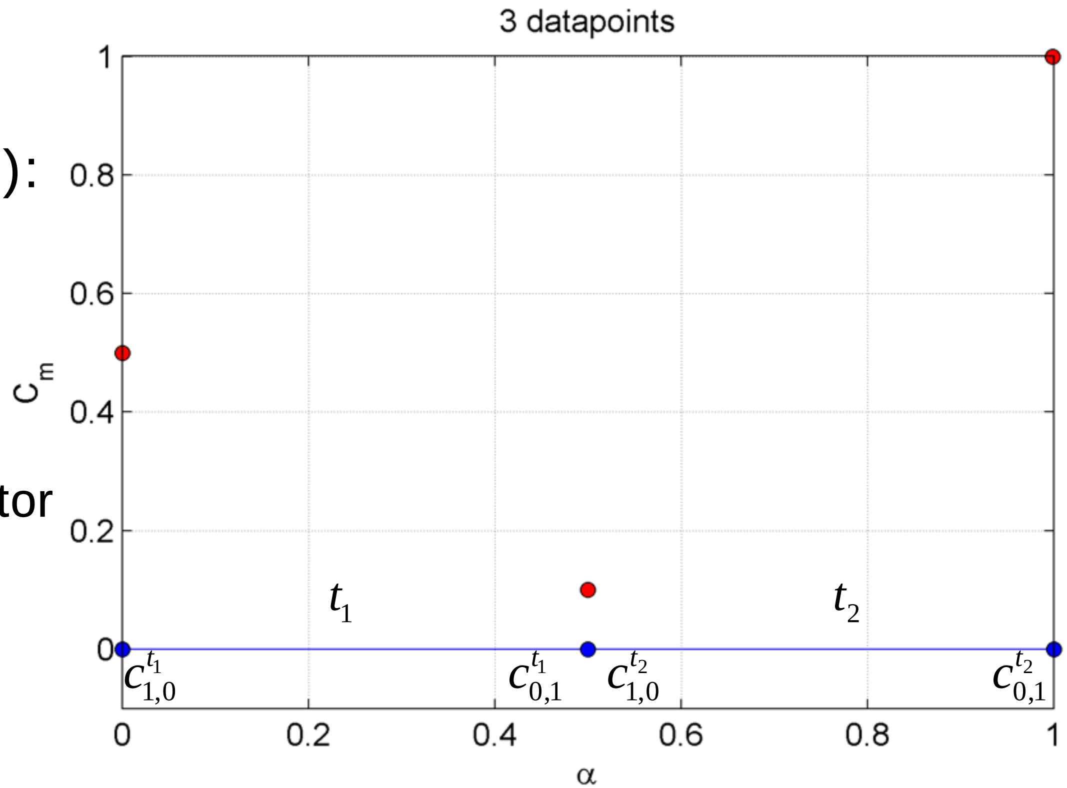
Example 8: 1-D data approximation with linear simplex splines

Step 5: Solve constrained linear regression problem (Lagrange Multiplier Method):

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} X & H^T \\ H & 0 \end{bmatrix} \begin{bmatrix} c \\ \lambda \end{bmatrix} + \varepsilon$$

→ Ordinary Least Squares Estimator

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} X & H^T \\ H & 0 \end{bmatrix}^{-1} \begin{bmatrix} X^T Y \\ 0 \end{bmatrix}$$

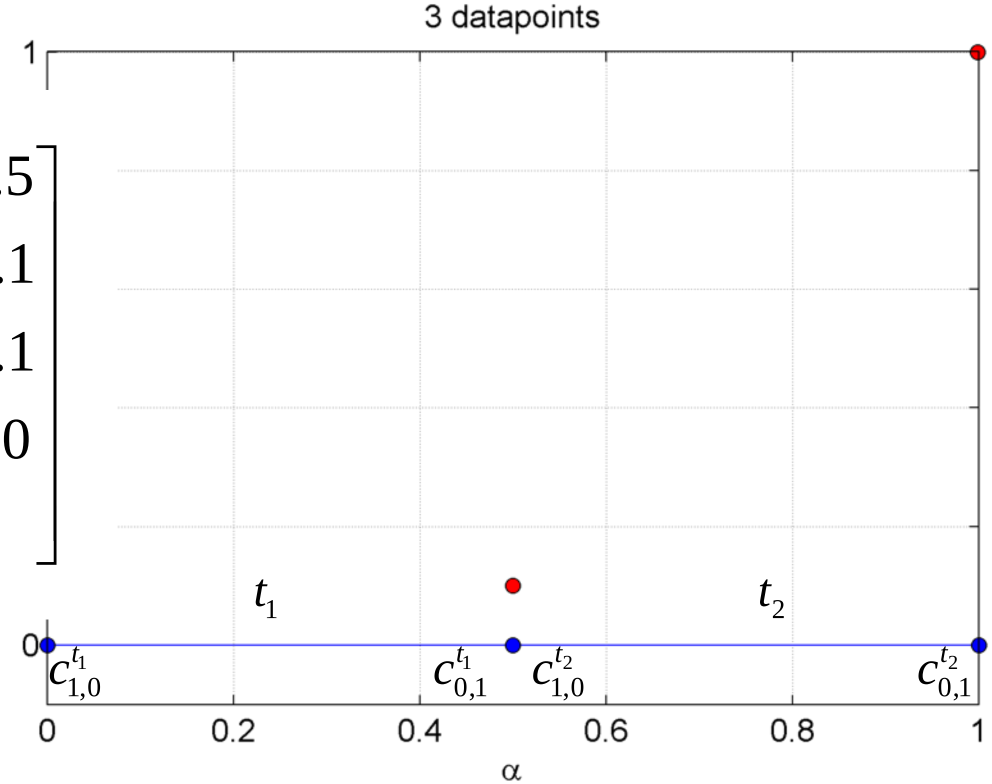


Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines

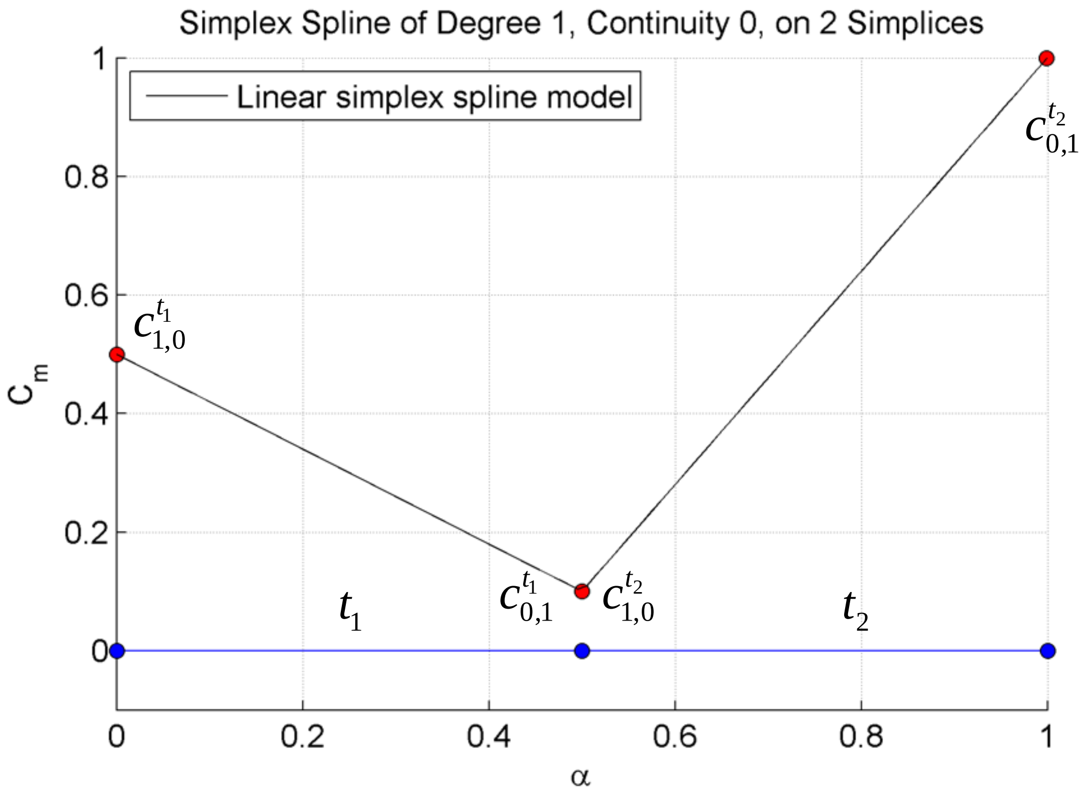
Step 5: Solved constrained linear regression problem:

$$\begin{bmatrix} c_{1,0}^{t_1} \\ c_{0,1}^{t_1} \\ c_{1,0}^{t_2} \\ c_{0,1}^{t_2} \\ \lambda \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & -0.5 \\ 0 & 0.5 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -0.5 & 0.5 & 0 & -0.5 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.1 \\ 0.1 \\ 1.0 \\ 0 \end{bmatrix}$$



Linear regression with simplex splines

Example 8: 1-D data approximation with linear simplex splines



Continuity of Simplex Splines

Example 6.5: First order continuity between two simplices; start with m=0

$$\boxed{c^{t_1}_{\kappa_0,m,\kappa_1}} = \sum_{|\gamma|=m} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_1), \quad m=0, d=4$$

<u>Left hand part</u> for m=0

Get all valid permutations of $(\kappa_0, 0, \kappa_1)$ with:

$$c^{t_1}_{\kappa_0,0,\kappa_1}$$

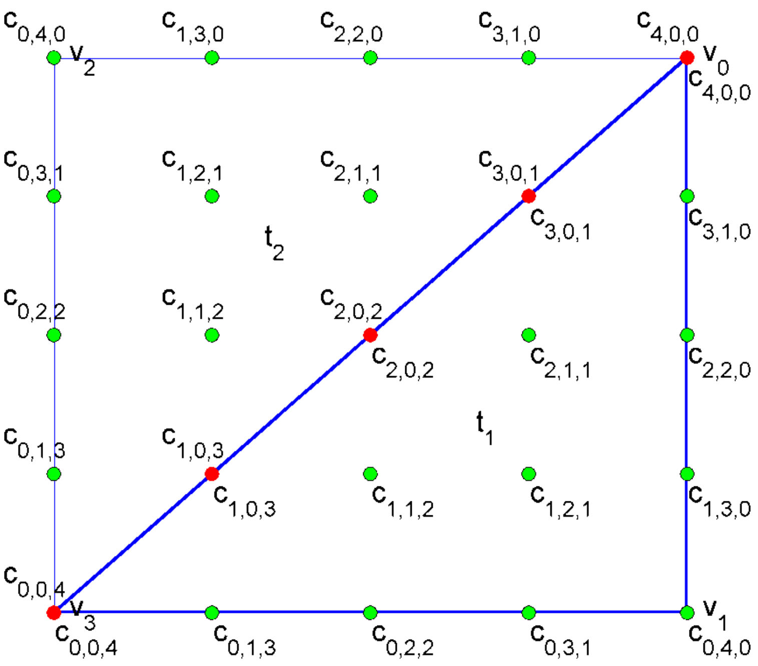
$$|(\kappa_0, 0, \kappa_1)| = 4$$

κ_0	0	κ_1
4	0	0
3	0	1
2	0	2
1	0	3
0	0	4

Rule:

$\kappa_0 + m + \kappa_1 = d$

$(\kappa_0 + 0 + \kappa_1) + (\gamma_0 + \gamma_1 + \gamma_2) = d$

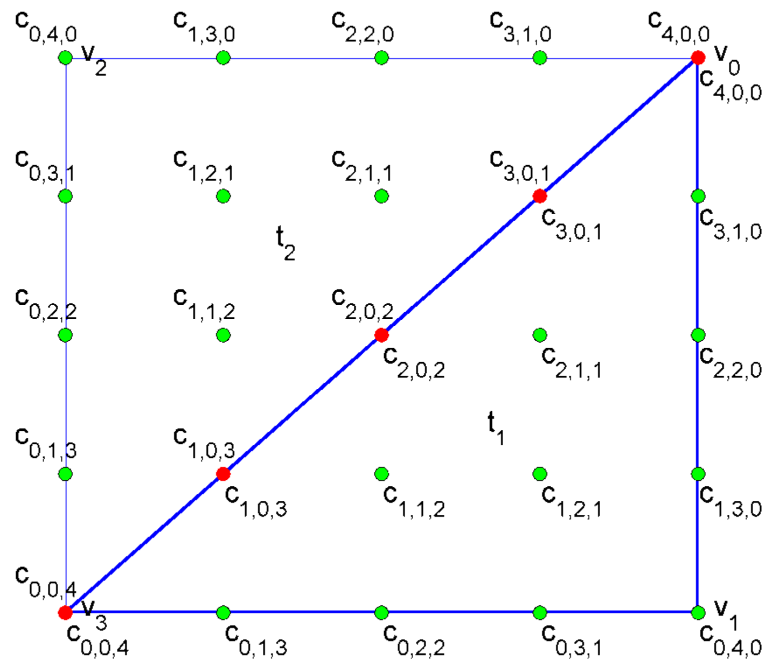


0th order continuity $(m = 0)$

Continuity of Simplex Splines

Example 6.5: First order continuity between two simplices; start with m=0

$$c^{t_1}_{\kappa_0,m,\kappa_1} = \sum_{|\gamma|=m} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_1), \quad m=0, d=4$$



0th order continuity $(m=0)$

<u>Right hand part</u> for m=0

Get all valid permutations of $(\kappa_0, 0, \kappa_1) + \gamma$ with:

$$|(\kappa_0, 0, \kappa_1) + \gamma| = 4, \ \gamma = \{(0,0,0)\}$$

$c^{t_1}_{\kappa_0,0,\kappa_1}$

κ_0	0	κ_1
4	0	0
3	0	1
2	0	2
1	0	3
0	0	4

$c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma}$

κ_0	0	κ_1	
4	0	0	$+(0,0,0)$
3	0	1	$+(0,0,0)$
2	0	2	$+(0,0,0)$
1	0	3	$+(0,0,0)$
0	0	4	$+(0,0,0)$

Continuity of Simplex Splines

Example 6.5: First order continuity between two simplices

$$\boxed{c^{t_1}_{\kappa_0,m,\kappa_1}} = \sum_{|\gamma|=m} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_1), \quad m=1, d=4$$

<u>Left hand part</u> for m=1

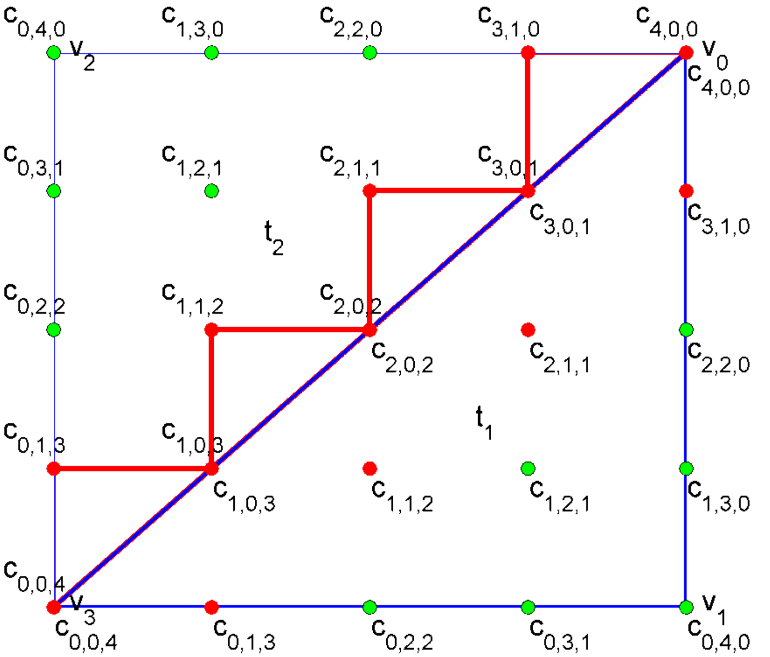
Get all valid permutations of $(\kappa_0, 1, \kappa_1)$ with:

$$|(\kappa_0, 1, \kappa_1)| = 4$$

$c^{t_1}_{\kappa_0,1,\kappa_1}$

κ_0	1	κ_1
3	1	0
2	1	1
1	1	2
0	1	3

Rule:
$\kappa_0 + m + \kappa_1 = d$
$(\kappa_0 + 0 + \kappa_1) + (\gamma_0 + \gamma_1 + \gamma_2) = d$



1st order continuity $(m = 1)$

Continuity of Simplex Splines

Example 6.5: First order continuity between two simplices

$$c^{t_1}_{\kappa_0,m,\kappa_1} = \boxed{\sum_{|\gamma|=m} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_1), \quad m=1}, d=4$$

<u>Right hand part</u> for m=1

Get all valid permutations of $(\kappa_0, 0, \kappa_1) + \gamma$ with:

$$\left|(\kappa_0, 0, \kappa_1) + \gamma\right| = 4, \ \gamma = \{(1,0,0),(0,1,0),(0,0,1)\}$$

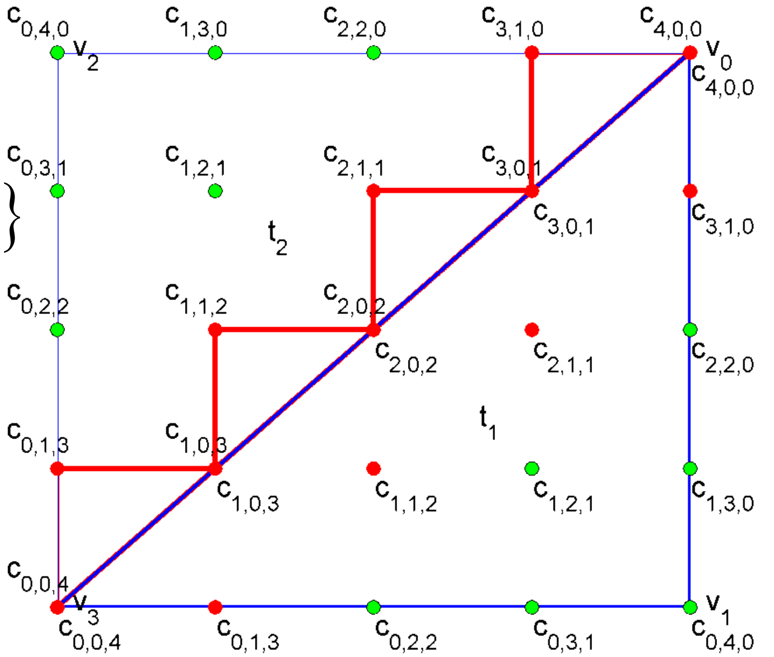
$c^{t_1}_{\kappa_0,1,\kappa_1}$

κ_0	1	κ_1
3	1	0
2	1	1
1	1	2
0	1	3

$\longrightarrow$

$c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma}$

κ_0	0	κ_1	
3	0	0	$+\gamma$
2	0	1	$+\gamma$
1	0	2	$+\gamma$
0	0	3	$+\gamma$



1st order continuity $(r = 1)$

Continuity of Simplex Splines

Example 6.5: First order continuity between two simplices

$$c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma}$$

$$\begin{array}{|c|c|c|}\hline \kappa_0 & 0 & \kappa_1 \\ \hline 2 & 0 & 1 \\ \hline \end{array} + \gamma = \begin{array}{|c|c|c|}\hline \kappa_0 & 0 & \kappa_1 \\ \hline 2 & 0 & 1 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline \gamma_0 & \gamma_1 & \gamma_2 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

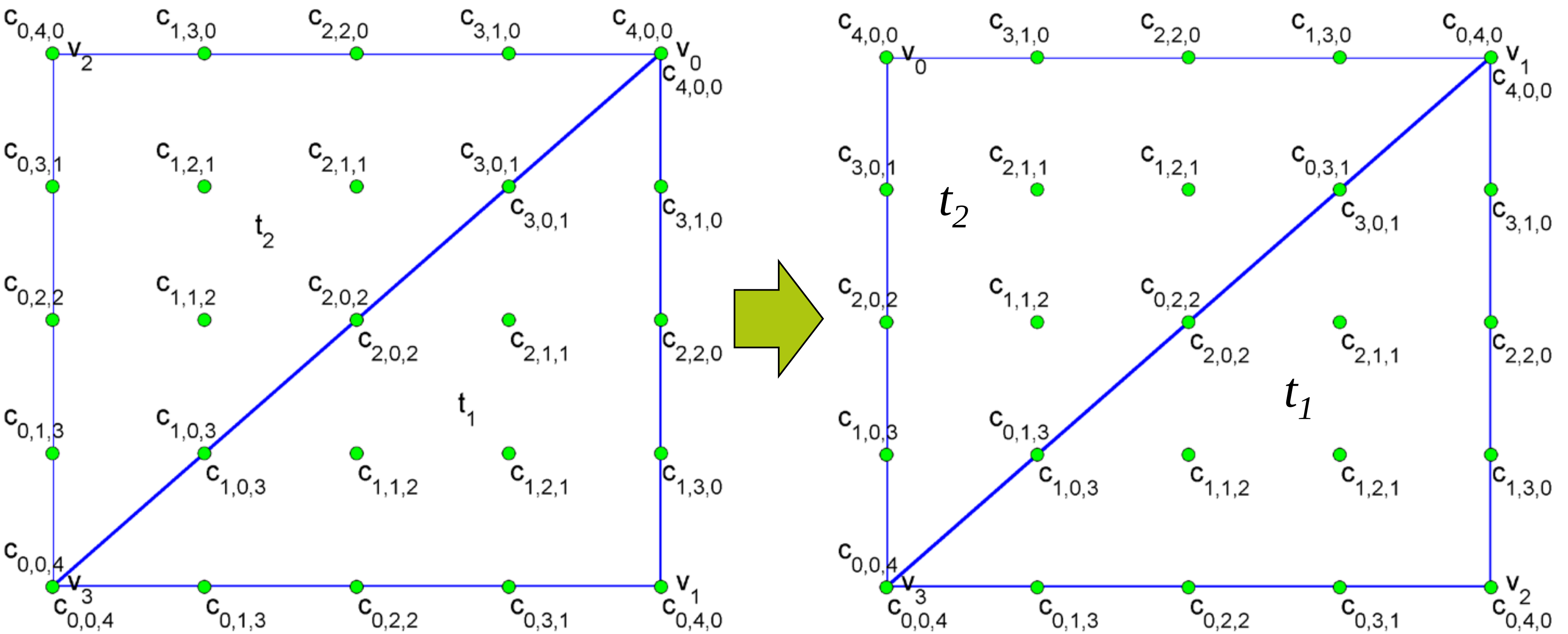
$$= \begin{array}{|c|c|c|}\hline \kappa_0+\gamma_0 & 0+\gamma_1 & \kappa_1+\gamma_2 \\ \hline 3 & 0 & 1 \\ \hline 2 & 1 & 1 \\ \hline 2 & 0 & 2 \\ \hline \end{array}$$

And so on…

Continuity of Simplex Splines

Example 6.6: Failed continuity between two simplices

We now change the B-net orientation such that the B-nets are no longer symmetric.



Continuity of Simplex Splines

Example 6.6: Failed continuity between two simplices

We now change the B-net orientation such that the B-nets are no longer symmetric. Now construct the continuity conditions for $r = m = 0$:

$$\cancel{c^{t_1}_{\kappa_0,0,\kappa_1} = \sum_{|\gamma|=0} c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma} B^0_\gamma(v_2)}$$

For $\gamma = (0,0,0)$ we have:

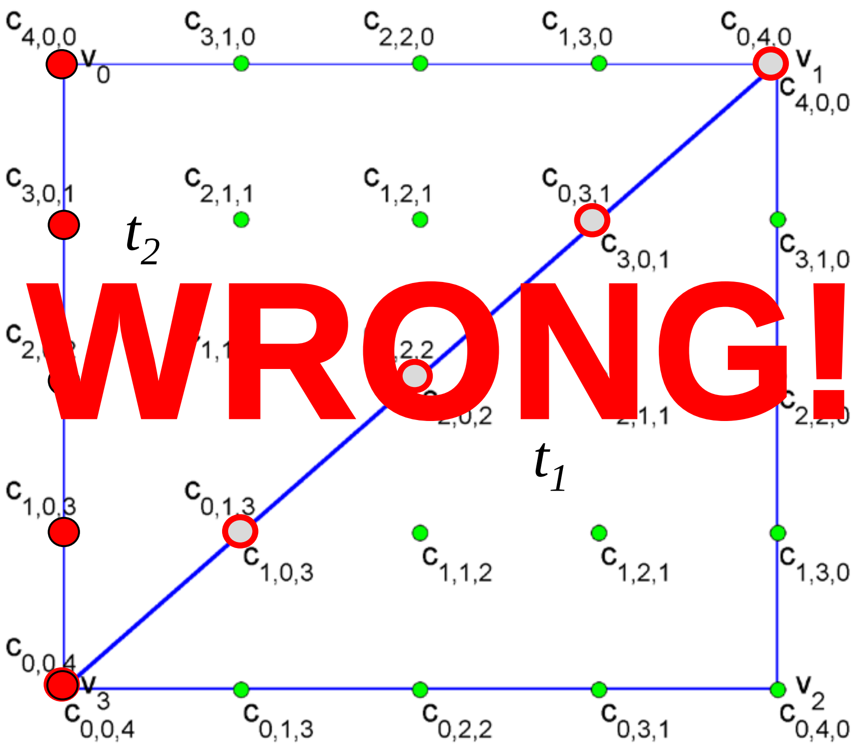
$$c^{t_1}_{\kappa_0,0,\kappa_2} = c^{t_2}_{(\kappa_0,0,\kappa_2)+(0,0,0)} B^0_{(0,0,0)}(v_2)$$

Get all valid permutations of $(\kappa_0, 0, \kappa_1)$:

$$(\kappa_0, 0, \kappa_1) \in \left\{ \begin{matrix} (4,0,0),(3,0,1), \\ (2,0,2),(1,0,3),(0,0,4) \end{matrix} \right\}$$

Get all valid permutations of $(\kappa_0, 0, \kappa_1) + \gamma$:

$$\cancel{(\kappa_0, 0, \kappa_1) + \gamma \in \left\{ \begin{matrix} (4,0,0),(3,0,1), \\ (2,0,2),(1,0,3),(0,0,4) \end{matrix} \right\}}$$



Continuity of Simplex Splines

Example 6.7: First order continuity between two simplices revisited

For 1st order continuity (r=1) we have:

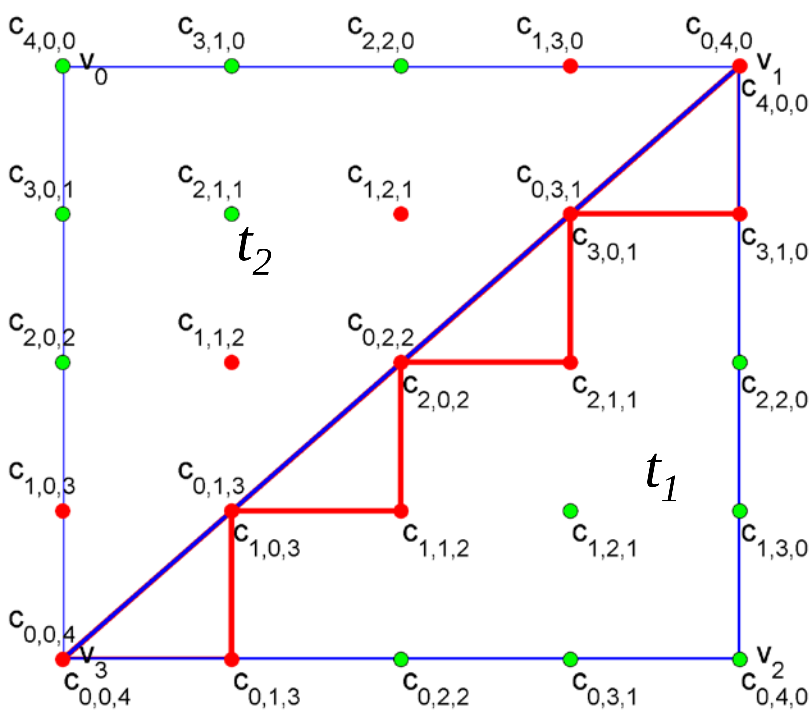
$$c^{t_2}_{m,\kappa_0,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_0), \quad 0 \le m \le 1$$

Note: we are now writing continuity from t_2 to t_1, which is always allowed!

Left hand part permutations (step 1):

Get all valid permutations of $\kappa = (1, \kappa_0, \kappa_1)$ with $|\kappa| = 4$:

$$\kappa = (1, \kappa_0, \kappa_1) \in \{(1,3,0), (1,2,1), (1,1,2), (1,0,3)\}$$



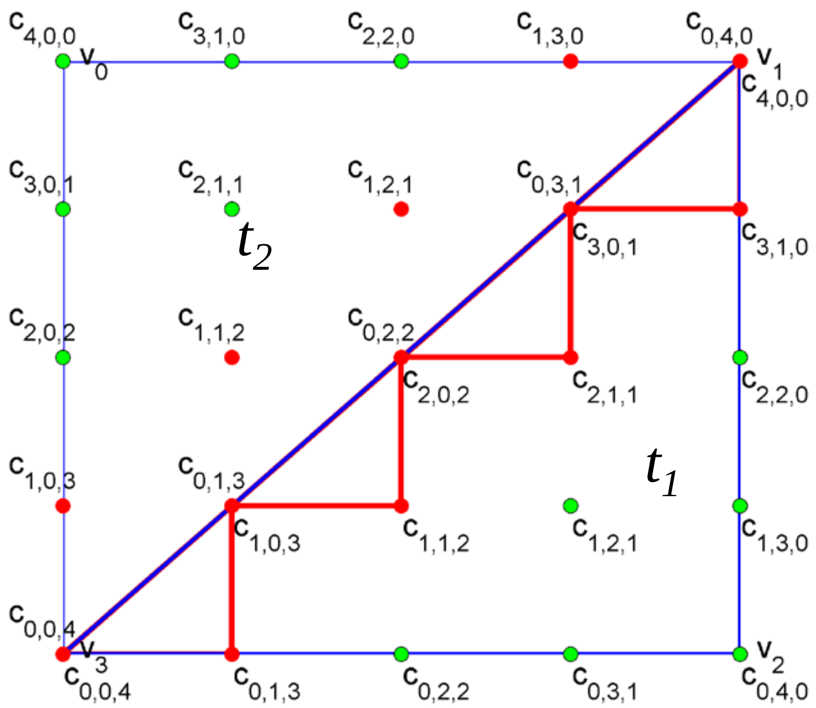
1st order continuity $(r = 1)$

Continuity of Simplex Splines

Example 6.7: First order continuity between two simplices revisited

For 1st order continuity we have:

$$c^{t_2}_{m,\kappa_0,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_0), \quad 0 \le m \le 1$$



1st order continuity $(r = 1)$

Right hand permutations (step 2):

Get all valid permutations of $\gamma = (\gamma_0, \gamma_1, \gamma_2)$ with $|\gamma| = 1$:

$$\gamma = (\gamma_0, \gamma_1, \gamma_2) \in \{(1,0,0),(0,1,0),(0,0,1)\}$$

Continuity of Simplex Splines

Example 6.7: First order continuity between two simplices revisited

$$c^{t_2}_{m,\kappa_0,\kappa_1} = \sum_{|\gamma|=m} c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma} B^m_\gamma(v_0), \quad 0 \le m \le 1$$

right hand permutations:

Get all valid permutations of $(\kappa_0, 0, \kappa_1) + \gamma$ with:

$$\left|(\kappa_0, 0, \kappa_1) + \gamma\right| = 4,\ \gamma = \{(1,0,0),(0,1,0),(0,0,1)\}$$

$c^{t_2}_{1,\kappa_0,\kappa_1}$

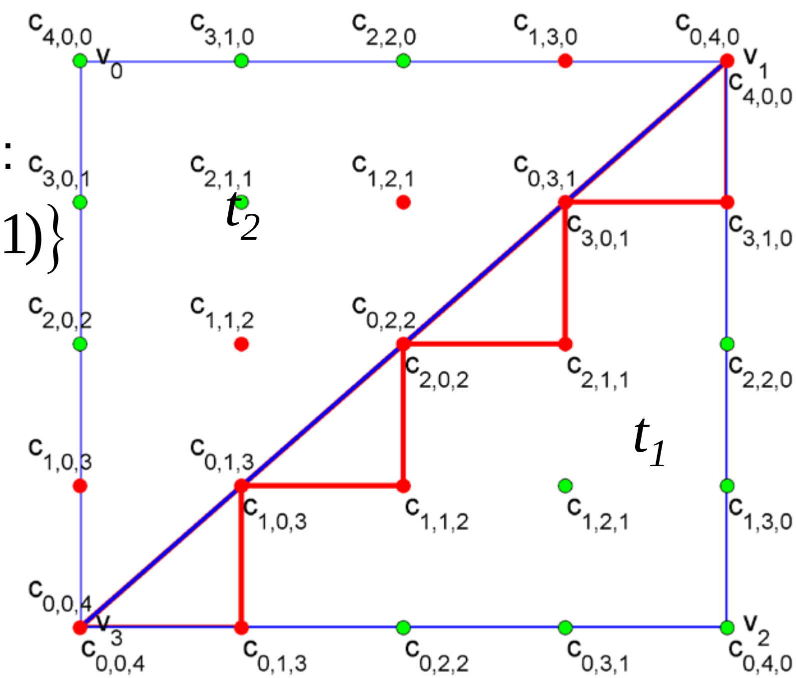
1	κ_0	κ_1
1	3	0
1	2	1
1	1	2
1	0	3

→

$c^{t_1}_{(\kappa_0,0,\kappa_1)+\gamma}$

κ_0	0	κ_1
3	0	0
2	0	1
1	0	2
0	0	3

$+\gamma$



1st order continuity $(r = 1)$

Continuity of Simplex Splines

Example 6.7: First order continuity between two simplices revisited

For example, for coefficient $c^{t_1}_{(3,0,0)+\gamma}$ we have:

$$c^{t_1}_{(3,0,0)+\gamma}$$

$$\begin{array}{|c|c|c|}\hline \kappa_0 & 0 & \kappa_1 \\ \hline 3 & 0 & 0 \\ \hline \end{array} + \gamma = \begin{array}{|c|c|c|}\hline \kappa_0 & 0 & \kappa_1 \\ \hline 3 & 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|}\hline \gamma_0 & \gamma_1 & \gamma_2 \\ \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array}$$

$$= \begin{array}{|c|c|c|}\hline \kappa_0+\gamma_0 & 0+\gamma_1 & \kappa_1+\gamma_2 \\ \hline 4 & 0 & 0 \\ \hline 3 & 1 & 0 \\ \hline 3 & 0 & 1 \\ \hline \end{array}$$

Continuity of Simplex Splines

Example 6.7: First order continuity between two simplices revisited

$c^{t_2}_{(\kappa_0,0,\kappa_1)+\gamma}$

κ_0	0	κ_1	
3	0	0	$+\gamma$
2	0	1	$+\gamma$
1	0	2	$+\gamma$
0	0	3	$+\gamma$

$=$

$\kappa_0+\gamma_0$	$0+\gamma_1$	$\kappa_1+\gamma_2$
4	0	0
3	1	0
3	0	1

$\kappa_0+\gamma_0$	$0+\gamma_1$	$\kappa_1+\gamma_2$
3	0	1
2	1	1
2	0	2

$\kappa_0+\gamma_0$	$0+\gamma_1$	$\kappa_1+\gamma_2$
2	0	2
1	1	2
1	0	3

$\kappa_0+\gamma_0$	$0+\gamma_1$	$\kappa_1+\gamma_2$
1	0	3
0	1	3
0	0	4