The Multivariate Multiplex B-Spline

AE4320: System Identification of Aerospace Vehicles

ir. Tim Visser

23-03-2015

Guest lecture of Monday March 23rd 2015, week 13 (or 3.7).

Personal details

Tim Visser

- MSc degree from Control and Simulation department in November 2014;
- PhD in Astrodynamics department since December 2014.

Contact information:

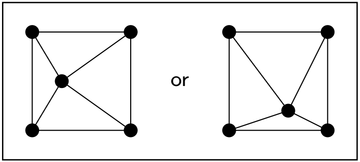
- Email: t.visser-1@tudelft.nl
- Office: HB 9.18

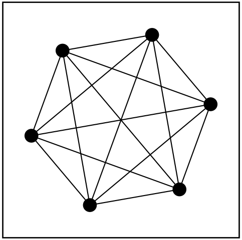
- You can always come by my office, no appointment required.
- I always try to respond to emails instantly.
- If you are looking for a thesis project, honors track assignment or a hobby, let me know.

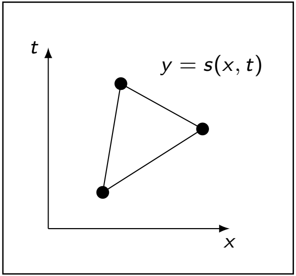
Motivation

Simplex splines have high approximation power and smoothness, but:

- How to triangulate?
- How to fill an *n*-simplex?
- How to include expert knowledge?







- Simplex splines as presented by C.C. de Visser are a strong modeling tool: high approximation power, flexibility in definition (degree, continuity order, triangulation (!)) and arbitrary smoothness.
- Three problems remain:
  1. There is no known (satisfactory) method of triangulating a domain in an optimal way.
  2. Simplices get increasingly hard to fill with data as the dimension increases.
  3. In the polynomials all high order terms and cross-couplings occur between all states.
- The multiplex spline can (partially) solve the last two problems.
- For the problem of triangulation, some advanced, yet computationally inefficient, yet again extremely interesting methods have been developed that use interval analysis. Contact Coen de Visser for more information.

Outline

- Introduction
- Bottom-up definition (BU)
  - Basis polynomials
  - B-net
  - Continuity
  - The multiplex
- Top-down definition (TD)
- Future research

- In the first hour we look at the multiplex spline from the bottom up.
- We will discuss all important spline aspects: basis polynomials, B-net, continuity.
- The most important part is the discussion of the multiplex itself, as it forms the starting point for the second hour.

BU: Origin

Solving the Hamilton-Jacobi-Bellman equation

$$\frac{\partial V(\boldsymbol{x},t)}{\partial t} + \inf_{\boldsymbol{u}} \left[\frac{\partial V(\boldsymbol{x},t)}{\partial \boldsymbol{x}} \cdot \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + r(\boldsymbol{x},\boldsymbol{u}) \right] = 0 \tag{1}$$

Use a spline to describe $V(\boldsymbol{x},t)$, 'split' dependency on $\boldsymbol{x}$ and t

$$\begin{aligned} V(\boldsymbol{x},t) &= \sum_{|\lambda_{\boldsymbol{x}}|=d_{\boldsymbol{x}}} c_{\lambda_{\boldsymbol{x}}}(t) B^{d_{\boldsymbol{x}}}_{\lambda_{\boldsymbol{x}}}(\boldsymbol{x}) \\ \text{with } c_{\lambda_{\boldsymbol{x}}}(t) &= \sum_{|\lambda_t|=d_t} c_{(\lambda_{\boldsymbol{x}},\lambda_t)} B^{d_t}_{\lambda_t}(t) \end{aligned} \tag{2}$$

- The concept of the multiplex spline was first introduced at this faculty by Govindarajan.
- In his method the B-coefficients of a simplex spline are described using a one-dimensional simplex spline in another dimension. He used this to separate state and time to simplify solving a partial differential equation (Hamilton-Jacobi-Bellman). These two subsplines are called the *layers* of the resulting spline. That is, this spline has a state layer and a time layer. For the remainder of this lecture we will look at a more general first and second layer.
- Note that I use λ for the multi-index κ. This is because we will need the simplex spline notation later.

BU: Basis Polynomials

Combine the expressions in $\boldsymbol{x}$ and t to find

$$\begin{aligned} V(\boldsymbol{x}, t) &= \sum_{|\lambda_{\boldsymbol{x}|}=d_{\boldsymbol{x}}} \sum_{|\lambda_t|=d_t} c_{(\lambda_{\boldsymbol{x}},\lambda_t)} B^{d_t}_{\lambda_t}(t) B^{d_{\boldsymbol{x}}}_{\lambda_{\boldsymbol{x}}}(\boldsymbol{x}) \\ &= \sum_{|\lambda_{\boldsymbol{x}|}=d_{\boldsymbol{x}},|\lambda_t|=d_t} c_{(\lambda_{\boldsymbol{x}},\lambda_t)} \mathcal{B}^{(d_{\boldsymbol{x}},d_t)}_{(\lambda_{\boldsymbol{x}},\lambda_t)}(\boldsymbol{x}, t) \end{aligned} \tag{3}$$

$$\text{with } \mathcal{B}^{(d_{\boldsymbol{x}},d_t)}_{(\lambda_{\boldsymbol{x}},\lambda_t)}(\boldsymbol{x}, t) = \frac{d_{\boldsymbol{x}}!}{\lambda_{\boldsymbol{x}}!} \frac{d_t!}{\lambda_t!} \beta^{\lambda_{\boldsymbol{x},0}}_{\boldsymbol{x},0} \cdot \ldots \cdot \beta^{\lambda_{\boldsymbol{x},n}}_{\boldsymbol{x},n} \cdot \beta^{\lambda_{t,0}}_{t,0} \cdot \beta^{\lambda_{t,1}}_{t,1}$$

Or in vector form

$$\boldsymbol{\mathcal{B}}^{d_{\boldsymbol{x}},d_t}_{n,1} = \boldsymbol{B}^{d_{\boldsymbol{x}}}_n \otimes \boldsymbol{B}^{d_t}_1 \tag{4}$$

- If we write out equation (2) we obtain a product of basis polynomials of different degree within the summations.
- The (simplex) basis polynomials can be combined to form the tensor-product Bernstein polynomials. The tensor-product is in the fact that each basis polynomial in $\boldsymbol{x}$ is combined with each polynomial in t. In vector form this is a tensor product.
- The tensor-product Bernstein basis polynomials are simply a product of the well known simplex basis polynomials.
- Note that I use β for the barycentric coordinates in the layers, again to distinguish them from the simplex barycentric coordinates.

BU: Basis Polynomials (2)

Let's compare the simplex and tensor-product polynomials

Simplex basis polynomials:

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

For example $n = 1$ and $d = 2$

$$\begin{bmatrix} b_0^2 & 2b_0 b_1 & b_1^2 \end{bmatrix}$$

Tensor-product basis polynomials:

$$\mathcal{B}_\lambda^d = \frac{d!}{\lambda!} \beta^\lambda$$

For example $\ell = 2$, $\nu = (1, 1)$ and $d = (2, 1)$

$$\begin{bmatrix} \beta_{10}^2 & 2\beta_{10}\beta_{11} & \beta_{11}^2 \end{bmatrix} \otimes \begin{bmatrix} \beta_{20} & \beta_{21} \end{bmatrix}$$
$$= \begin{bmatrix} \beta_{10}^2\beta_{20} & \beta_{10}^2\beta_{21} \\ 2\beta_{10}\beta_{11}\beta_{20} & 2\beta_{10}\beta_{11}\beta_{21} \\ \beta_{11}^2\beta_{20} & \beta_{11}^2\beta_{21} \end{bmatrix}$$

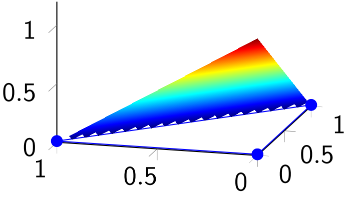
- It is important to realize what this different form of polynomial implies for your model. It assumes a different relation between the variables of your model. One might argue that the tensor-product polynomials ensure a decoupling between the layers. After all, the polynomial in one layer is completely independent from the polynomials in other layers.
- Please note there is a difference between the two occurrences of d on this slide. On the left it is just a single value, whereas on the right it is a collection (in a vector) of all degrees in the different layers.
- The collection of basis polynomials on this slide should not be seen as a matrix with two columns. In reality this tensor-product will result in a row vector. I just did not have space to put them all side by side.

BU: Basis Polynomials (3)

Let's compare the simplex and tensor-product polynomials

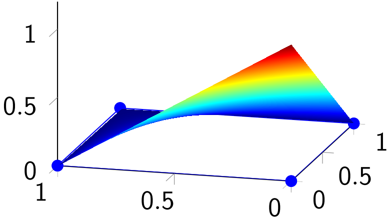
Simplex basis polynomials:

$$B^1_{100} = b_0$$



Tensor-product basis polynomials:

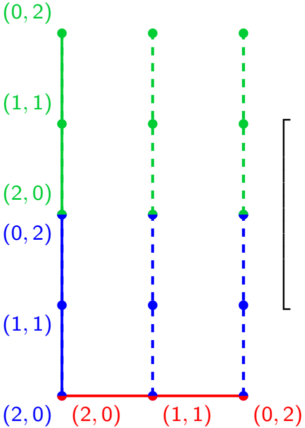
$$\mathcal{B}^{(1,1)}_{1010} = \beta_{10}\beta_{20}$$



- As another example, let's look at the linear basis polynomials in both cases. On the left we have the linear basis polynomial on a simplex. We can clearly see that this is a simple flat surface. This does not surprise us: the basis polynomial is as linear as possible.
- On the right we see a so called bilinear basis polynomial. We can clearly see that this function does not form a straight surface, but it has a certain curve to it. This does not surprise us either, as the basis polynomial is a product between two variables, so it is actually a parabola-like function.
- Note that both these basis functions are the simplest possible in their field. That is, you cannot have a more linear tensor-product basis polynomial in two dimensions than the one shown on this slide. This means that the actual degree of these basis polynomials can never be lower than ℓ, the number of layers. After all, you need at least degree 1 in each layer.
- We can illustrate the effect of using bilinear functions by considering the diagonal: $b_{10} = b_{20}$. This yields b_{10}^2. That is, along the diagonal we have a parabola. The degree we see along the diagonal of the basis polynomials is called the *total degree*, and it is defined as the sum of the degrees in the layers: $|d|$.

BU: B-net

Each combination of λ_1 and λ_2 occurs in the B-net:

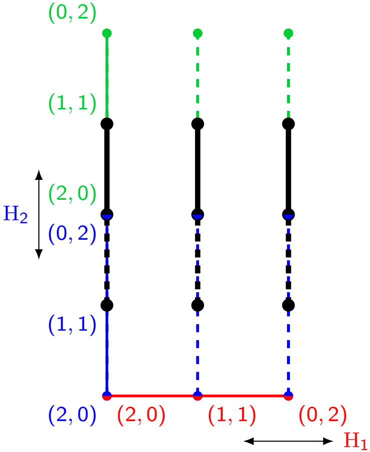


$$\begin{bmatrix} (2,0,2,0) \\ \vdots \\ (2,0,0,2) \\ (1,1,2,0) \\ \vdots \\ (0,2,0,2) \end{bmatrix}$$

- The B-net of the tensor-product spline can be deduced from the idea of describing the B-coefficients of one spline with a spline in another variable/dimension. For each coefficient in one layer we get a copy of the entire B-net of the second. This is an iterative process over all layers.
- As an example we look at the biquadratic case ($d = (2, 2)$). Note how the B-coefficients form a set of parallel lines.
- The resulting list of multi-index permutations (or B-coefficients) takes the order depicted in this slide. First all coefficients with first entries (2,0) from the first layer are listed in lexicographical order, then those with (1,1) at the start and finally those with (0,2) in the first layer.
- As a nice trick: note that you can immediately spot the polynomial degree from a B-net, independent of the dimension of the spline. Just count the number of coefficients along an edge (line between two vertices) and subtract 1. In a multiplex this trick also works, but one should be careful to choose an edge in the layer of interest. Of course this trick has an origin in the definition of the polynomials. Can you derive it?

BU: Continuity conditions

Every copy of the B-net gets a copy of the continuity conditions



$$\mathrm{H} = \begin{bmatrix} \mathrm{H}_1 \otimes \mathbb{I}_{N_2 \hat{d}_2} \\ \mathbb{I}_{N_1 \hat{d}_1} \otimes \mathrm{H}_2 \end{bmatrix}$$

- A small note on continuity conditions in the multiplex spline. Like the B-net, the continuity conditions derived from the different layers should be copied to hold for all copies of the B-net.
- Copying a matrix is done using the tensor-product with an appropriately constructed matrix. In the discussed approach with two layers, these are identity matrices. When more layers are used, we can think of an iterative algorithm in which we first construct H for the first two layers and then use it as H_1 in the same expression.
- Note that the coloring in this slide refers to the layers. Red is the direction in which the red simplex lies, blue the direction of the blue one. However, in the cases displayed in the these slides H_1 would be empty (because there is only one simplex, so no continuity between simplices) and H_2 contains the conditions between the blue and the green simplex.
- In the dimension of the identity matrices, N_i is the amount of simplices in layer i and $\hat{d}_i$ is the amount of permutations of the multi-index λ_i.

BU: Summary

In general, we may use ℓ *layers* of arbitrary dimensions $\nu = (\nu_1, ..., \nu_\ell)$

$$\pi(\boldsymbol{x}) = \sum_{|\lambda|_\nu = d} c_\lambda \mathcal{B}^d_\lambda(\boldsymbol{x})$$
$$\mathrm{H}^\ell \boldsymbol{c} = \begin{bmatrix} \mathrm{H}^{\ell-1} \otimes \mathbb{I} \\ \mathbb{I} \otimes \mathrm{H}_\ell \end{bmatrix} \boldsymbol{c} = \mathbf{0} \tag{5}$$

for which we introduce:

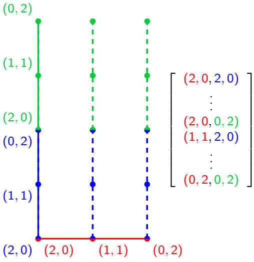
- multi-index $\lambda = (\lambda_1, ..., \lambda_\ell)$
- degrees in a vector $d = (d_1, ..., d_\ell)$
- barycentric coordinates $\beta = (\beta_1, ..., \beta_\ell)$
- partial norm $|\lambda|_\nu = (|\lambda_1|, ..., |\lambda_\ell|)$
- basis polynomials $\mathcal{B}^d_\lambda(\boldsymbol{x}) = \frac{d!}{\lambda!}\beta^\lambda$

- In more general terms we can extent the tensor-product to include more layers of arbitrary dimension and degree.
- To arrive at a concise notation we introduce
  - the collection of multi-indices from all the layers λ
  - a vector containing all the degrees in the different layers d
  - the collection of barycentric coordinates β
  - a partial norm $|\lambda|_\nu$ that allows for a short way of writing $|\lambda_i| = d_i, \forall i \in [1, \ell]$
  - basis polynomials $\mathcal{B}^d_\lambda(\boldsymbol{x})$ that are then completely the same in written form as the simplex spline polynomials. Note however that this time $d!$ is a factorial of a vector, and therefore $d! = d_1! \cdot \ldots \cdot d_\ell!$ just like for the multi-index λ.
- Using this notation we find that we can write the fitting problem in exactly the same way as for the simplex spline. Therefore we can conclude that we can use all approximation methods described for those splines in earlier lectures.

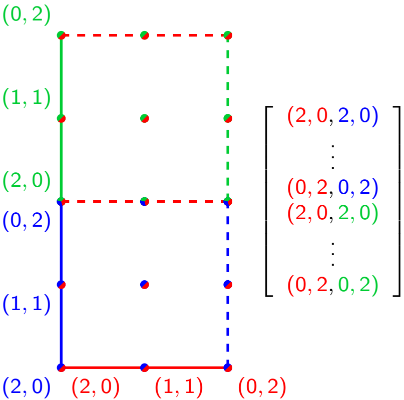
BU: B-net (2)

We can identify a second approach to describe the B-net

Splineception



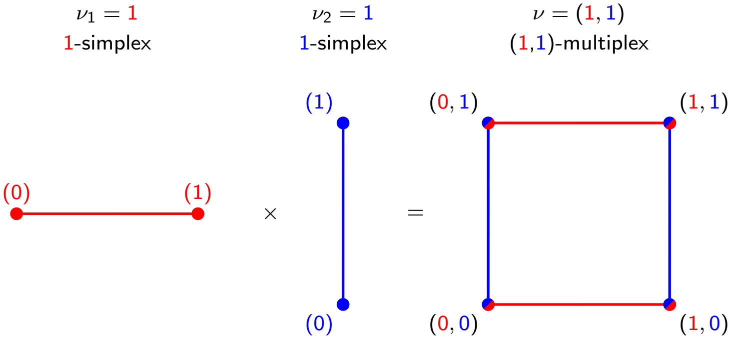
Multiplex spline



- From observing the B-net we can define the tensor-product spline in two ways. Both ways are equivalent. The only difference is the ordering of the B-coefficients.
- The first option is here referred to as "Splineception", which we have seen in the previous slides. Each B-coefficient of one spline is described using a spline in another dimension. Because the entire triangulation of the second spline is used for each coefficient of the first spline, we end up with the indicated ordering. In words: there is a complete spline behind each coefficient.
- The second option is (finally!) the multiplex spline. Here the coefficients are grouped per combination of simplices from each layer. That is, first we have all coefficients that lie in the first simplex in both layers, then the coefficients that lie in the first simplex in the first layer and the second simplex in the second layer, etcetera. By grouping the coefficients in this way, we arrive at the concept of a multiplex: a combination of low-dimensional simplices in different dimensions.

BU: Multiplex

The Cartesian product ($\times$) can be applied to find the multiplex

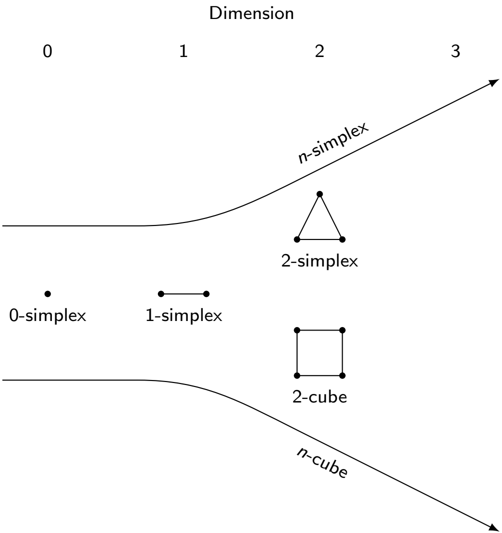


- We have seen that the B-net of the multiplex spline (or splineception approach) is formed by defining a B-coefficient at each 'combined' location of coefficients in the layers. This combination is called a Cartesian product in mathematics.
- The geometric basis on which the B-net lies and the spline is defined, is constructed in a similar way as the B-net: through a Cartesian product of the layers. As an example we look at the multiplex of a spline with the dimensions $\nu = (1,1)$. We have two layers with two vertices each, for example:

  - $v_{10} = 0$; $v_{11} = 1$
  - $v_{20} = 0$; $v_{21} = 1$

  Then the Cartesian product is each combination of the above vertices. Note that both layers are one-dimensional and perpendicular, so v_{1j} lie on the x-axis, whereas v_{2j} lie on the y-axis. Then $\mathcal{V}_1 \times \mathcal{V}_2 = \{(0,0),(0,1),(1,0),(1,1)\}$.
- Because the dimensions determine the overall shape of the multiplex, we often include them in the name: the *ν-multiplex*. The example given here is thus the $(1,1)$-multiplex.
- As a final note: the Cartesian product only works if the layers are in the same direction as the axis system you use to define the vertices.

BU: Multiplex (2)

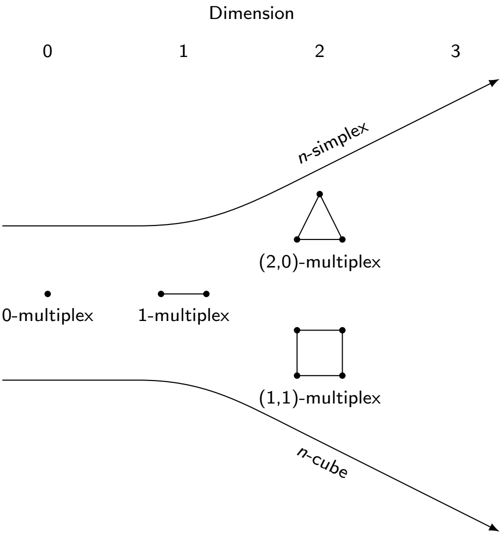


$$(b_0, b_1, b_2)$$

$$(b_0, b_1)$$

$$(\beta_{10}, \beta_{11}), (\beta_{20}, \beta_{21})$$

BU: Multiplex (2)

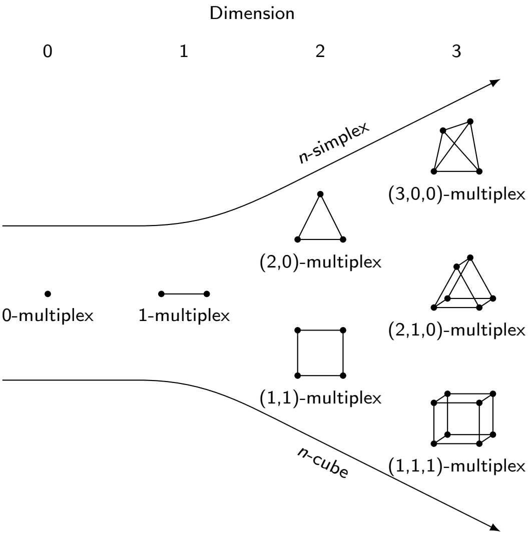


$$(\beta_{10}, \beta_{11}, \beta_{12}), (\beta_{20})$$

$$(\beta_{10}, \beta_{11})$$

$$(\beta_{10}, \beta_{11}), (\beta_{20}, \beta_{21})$$

BU: Multiplex (2)



$$(\beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}), (\beta_{20}), (\beta_{30})$$

$$(\beta_{10}, \beta_{11}, \beta_{12}), (\beta_{20}, \beta_{21}), (\beta_{30})$$

$$(\beta_{10}, \beta_{11}), (\beta_{20}, \beta_{21}), (\beta_{30}, \beta_{31})$$

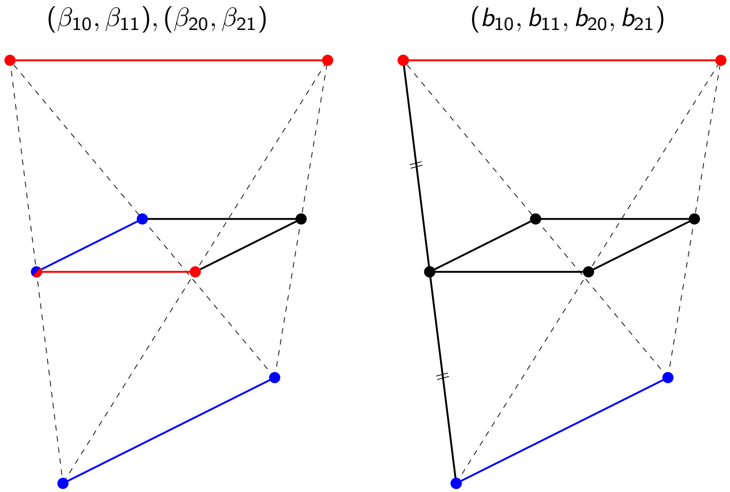
- The concept of taking a Cartesian product of simplices can be extended to any number of layers of any dimension. Even the simplices themselves can be seen as special cases of multiplices.
- Some notes regarding the overview of the multiplices presented in this slide:
  - With increasing dimension there are ever more different multiplices possible. No direct method exists to compute the amount of multiplex types for a given dimension n. This is the so called *integer partition problem*, which can only be solved using a generating function.
  - Along the top of the figure the simplices are positioned. Note that for the sake of generality zero-dimensional layers are added in the notation. Can you show that these layers do not change the shape of the multiplex? And that they do not affect the basis polynomials? (Hint: what is the value of β_{20} in the (2,0)-multiplex?)
  - Along the bottom of the figure the cubes are positioned. These are multiplices constructed using only one-dimensional layers. Note that these multiplices have n one-dimensional layers.
  - In between the simplex and the cube lie more and more multiplices as the dimension increases. In the most general sense one can arrive at each multiplex from an $(n-1)$-multiplex by increasing the dimension of one layer by one. There are $\tilde{\ell}$ ways to do this, if $\tilde{\ell}$ is the number of non-zero-, unique-dimensional layers of the n-dimensional multiplex. For example: the (1,1,1)-multiplex has only one unique layer (all three are the same), so we can only get there from the (1,1)-multiplex. The same holds for the (2,2,0,0)-multiplex, for example (accessible through the (2,1,0)-multiplex only). The (2,1,1,0)-multiplex however can be made by increasing the dimension in the first or second layer, from the (1,1,1)- or the (2,1,0)-multiplex respectively.
  - In the tree for lower dimensions we can easily identify the face that can be shared between two multiplices. If two multiplices can be constructed as described in the previous point from the same lower-dimensional multiplex, they can share a face in the shape of that multiplex. Very simplex example: both the triangle and the rectangle are derived from the line segment, so they can share a line segment.
  - We generally list the multiplices in the same order as the multi-index κ. Note however that it is not necessary to include all permutations: the order of the layers is not important.
  - To visualize the complete set we would need a multi-dimensional tree for high-dimensional multiplices, but with this drawing in two dimensions we are at least fine up to $n = 5$. More advanced drawings are for example the Young diagrams (see e.g. Wikipedia, integer partition).
- Let's do a quick quiz: how many four-dimensional multiplices are there? Write down the distributions of dimension (ν's). Can you draw them?

Outline

- Introduction
- Bottom-up definition (BU)
- Top-down definition (TD)
  - The circumscribed simplex
  - Basis polynomials
  - B-net
  - Continuity
- Future research

- In the second hour we will discuss the top down definition of the multiplex.
- Before we were combining lower dimensional layers to construct a multiplex, and the whole spline. Now we will derive a lower dimensional multiplex and spline from a higher dimensional simplex spline.
- Again all the spline aspects will be discussed: geometry, basis polynomials and B-net and continuity conditions. This time however the order and focus is different. This time the spline structure is derived from the geometric point of view, and special attention is given to continuity conditions.

TD: Multiplex

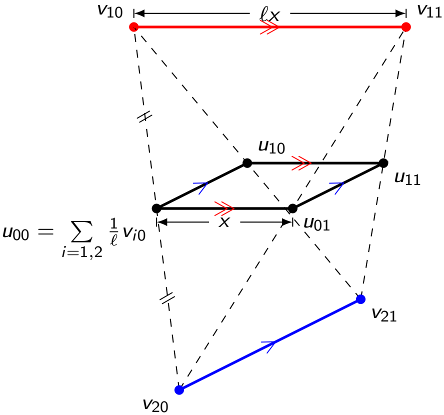


- Consider as an example the (1,1)-multiplex. We have seen before that we need two barycentric coordinates to describe the position of a point in each layer, so four in total. The pairs of coordinates sum to one, so summing all coordinates yields 2 at all times.
- Four barycentric coordinates also describe a location in the 3-simplex. What area is described in the simplex by the range of barycentric coordinates (scaled with 2)?
- Note that we have $b_{10} + b_{11} = \frac{1}{2}$ in the simplex, so we have to split the vertices in groups again. In the simplex we identify *layer sets* corresponding to the layers of the multiplex.
- Between vertices of the layer sets we can draw *simplex links*. These are $(\ell - 1)$-simplices that connect all layers. At the barycenter of these simplex links we find points that lie in the plane of interest.
- We find that the area in the simplex is again the multiplex. This leads to the *top-down definition* of the multiplex, and its *circumscribed simplex*.

TD: Multiplex (2)

In general the simplex and multiplex are related as follows:

- Circumscribed simplex dimension $= n + \ell - 1$
- Layers $\parallel$ layer sets
- Equidistant from layer sets
- Layer sets are scaled with ℓ
- Barycentric coordinates $b = \frac{1}{\ell}\beta$



- The top-down definition of the multiplex introduces many parallels between the multiplex and the simplex spline. To facilitate these derivations, it is useful to note the following parallels between the multiplex and its circumscribed simplex.
  - First note that we introduce a new name for the vertices of the multiplex: u_{ij}. Here i signifies the index of the vertex taken from the first layer, and j the index of the vertex from the second layer. That is, u_{ij} lies halfway the line between v_{1i} and v_{2j}.
  - The circumscribed simplex is of dimension $n + \ell - 1$. We can add zero dimensional layers at will.
  - A layer set is a parallel copy of the layer. In fact, each layer set is a pure translation (and scaling, see next point) of a layer.
  - The multiplex is positioned such that each vertex is equidistant from all defining vertices of the circumscribed simplex. In this case ($\ell = 2$), this means the vertices lie halfway the line between two vertices from different layer sets. If there are more layers, the vertices lie at the barycenter of the vertices from the different layer sets. That is $u_{\phi} = \sum_{i=1}^{\ell} \frac{1}{\ell} v_{i\phi_i}$, with ϕ a vector of the indices of vertices used from the different layer sets. Note how we use exactly one vertex from each layer set.
  - The layer set is scaled with ℓ compared to the layer. Note that ℓ includes any zero dimensional layers. That is, if you increase the dimension of the circumscribed simplex, the inscribed multiplex will become smaller.
  - Barycentric coordinates belong to a layer-layer set pair and relate as follows $b = \frac{1}{\ell}\beta$.
  - For automation purposes it is useful to define a standard algorithm for constructing the circumscribed simplex. We currently use the following procedure:
    1. Choose the layers of the multiplex;
    2. Scale all vertex locations with ℓ;
    3. For all layers $i \in [1,\, \ell - 1]$, append the unit vector $e_i \in \mathbb{R}^{\ell-1}$, where $e_{ij} = 1$ if $i = j$ and 0 otherwise, to the scaled vertex locations;
    4. In layer ℓ, append $-\mathbf{1}_{\ell-1} = (-1,\, \ldots,\, -1) \in \mathbb{R}^{\ell-1}$ to the scaled vertex locations.

TD: Basis polynomials

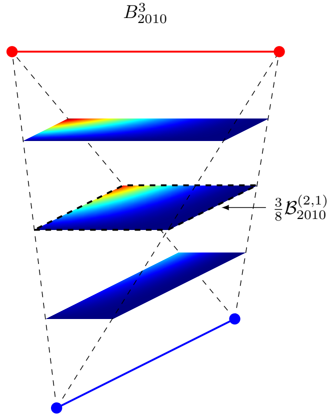
κ for $|d| = 3$ $\qquad\qquad$ λ for $d = (2,1)$

$$B_\lambda^{|d|} = \frac{|d|!}{\lambda!} b^\lambda \qquad \begin{bmatrix} (3,0,0,0) \\ (2,1,0,0) \\ (2,0,1,0) \\ (2,0,0,1) \\ (1,2,0,0) \\ (1,1,1,0) \\ (1,1,0,1) \\ (1,0,2,0) \\ (1,0,1,1) \\ (1,0,0,2) \\ (0,3,0,0) \\ (0,2,1,0) \\ (0,2,0,1) \\ (0,1,1,1) \\ (0,0,3,0) \\ (0,0,2,1) \\ (0,0,1,2) \\ (0,0,0,3) \end{bmatrix} \xrightarrow{\mathcal{B}_\lambda^d = \ell^{|d|} \frac{d!}{|d|!} B_\lambda^{|d|}} \begin{bmatrix} (2,0),(1,0) \\ (2,0),(0,1) \\ (1,1),(1,0) \\ (1,1),(0,1) \\ (0,2),(1,0) \\ (0,2),(0,1) \end{bmatrix} \qquad \mathcal{B}_\lambda^d = \frac{d!}{\lambda!}\beta^\lambda$$

Multiplex basis polynomials are a scaled subset of higher dimensional, total degree simplex polynomials.

- Like the multiplex, the tensor-product basis polynomials can also be found in the circumscribed simplex. Note that, apart from some scaling, the tensor-product polynomials are elements from the set of basis polynomials of a simplex spline of dimension $n + \ell - 1$ and total degree $|d|$. The whole set of multiplex basis polynomials can be found in the set of simplex basis polynomials.
- The scaling of the basis polynomials is due to the multinomial coefficients and the barycentric coordinates. Can you derive the scaling factor?

TD: Basis polynomials (2)

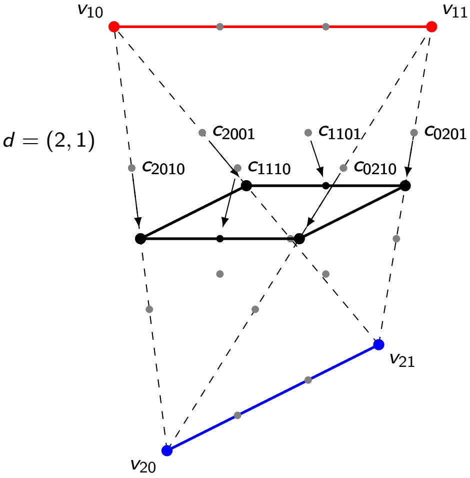


- Here we see a basis polynomial (B^3_{2010}) in the 3-simplex, illustrated using three two-dimensional slices. This basis polynomial has its maximum somewhere between the top and the middle slice.
- By using the multiplex as a basis and degree $d = (2, 1)$, we get the top slice as a tensor-product basis polynomial. To make sure the maximum on this slice is of appropriate size, we scale the basis polynomial (as described in the previous slide). One can think of this scaling as a way to get the maximum value of the simplex basis polynomial to the location of the multiplex. At least it is there to guarantee the partition of unity property of the tensor-product basis polynomials.

TD: B-net

The B-net can be found in the circumscribed simplex, just like the basis polynomials

- Start with the simplex B-net
- Choose the degree in the layers
- Coefficients are 'projected' onto the multiplex

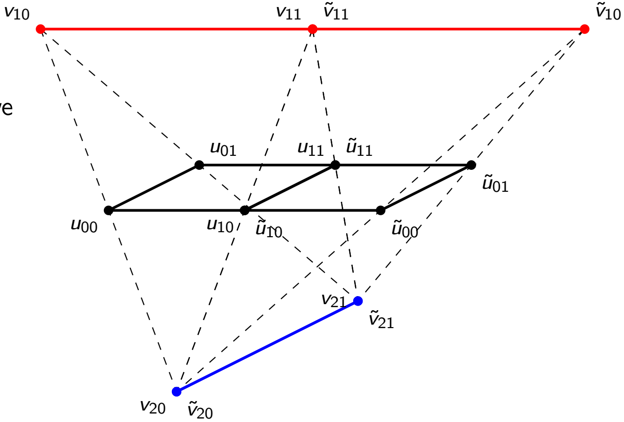


- Much in the same way as the basis polynomials, the B-coefficients can be found in the simplex spline. They are located on a slice of the B-net parallel to the multiplex, at the location where we also find the maxima of the relevant basis polynomials.
- With a given total degree, we can split up this degree in multiple ways over the layers. The scaling of the basis polynomials can be thought of as to position the coefficients onto the multiplex.
- Note that it is not generally true that the slice of the simplex B-net lies in the plane of the multiplex. For odd total degree this is not even possible. In other words, the location of the slice of the B-net depends only on the distribution of the total degree over the layers. This slice is however always parallel to the multiplex. To formalize this, we may define a *biased multiplex* as a multiplex that does not lie in a plane equidistant from the layer sets. Instead it 'favors' one layer set over the other, and lies at location for which $\sum_j b_{1j} = \alpha$ and $\sum_j b_{2j} = 1 - \alpha$, with α not necessarily $\frac{1}{2}$. Then the multiplex B-net lies on a biased multiplex with $\alpha = \frac{d_1}{|d|}$.
- Remember the trick to spot the degree of a spline by its B-net: count the B-coefficients on an edge and subtract one.

TD: Continuity

For continuity conditions we need to know the shared edge.

- Shared edge and out-of-edge layer;
- Circumscribed simplex for first multiplex;
- Circumscribed simplex for second multiplex;
- Shared simplex is circumscribed around shared edge.

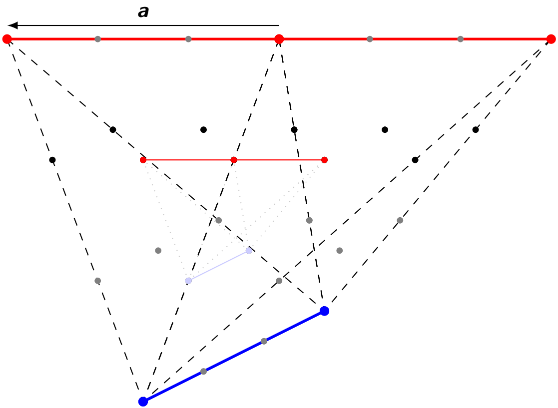


- The first step towards continuity between multiplices from the circumscribed simplex spline lies in the shared edge. We need to show that if the multiplices share an edge, we can construct simplices that also share an edge.
- The shared edge is simply the edge that is part of both multiplices. In this case this is the line between u_{10} and u_{11}. The out-of-edge layer is therefore the line u_{00} to u_{10}. This is the only layer that is not shared completely.
- It can be derived that the circumscribed simplex leaves enough freedom to define two simplices that also share an edge. The procedure would be to construct the simplex for one multiplex and then search for the right location for the out-of-edge vertex of the other simplex. The latter problem has a unique solution.
- Note that all layer sets are shared that belong to layers on the shared edge. Also, there is only one out-of-edge vertex for the simplices, even though there are multiple vertices of the multiplices that are not shared. However, we define the out-of-edge vertex of the multiplex as the vertex in the out-of-edge layer that corresponds to the out-of-edge vertex of the circumscribed simplex.

TD: Continuity (2)

Continuity conditions are drawn in the B-net.

- Isolate the multiplex B-net;
- Conditions are miniature simplices;
- Simplify to miniature layers.

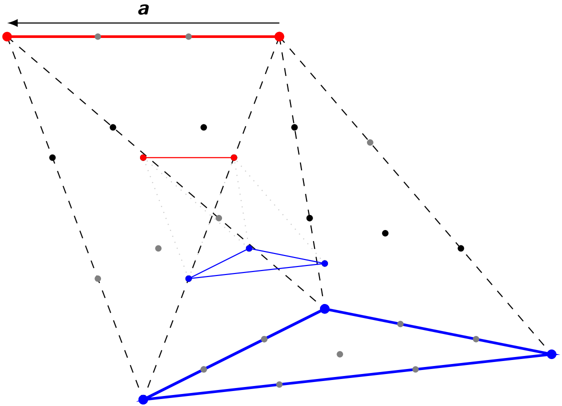


- Assuming we use the indicated B-net, we only have to worry about the indicated continuity conditions. Any conditions that do not use coefficients on the plane are trivially met, as all coefficients are zero out of the plane.
- Due to the direction in which the derivative is taken, the conditions simplify even more, such that we are left with only those coefficients that form a copy of the *out-of-edge layer*, or the layer in which the out-of-edge vertex lies.
- It is important to realize that this simplification is due to the fact that the red layer sets are in line, and not because many of the coefficients are zero. After all, if we would only use the latter fact, we could also find conditions where only the blue line lies in the B-net of the multiplex. This would result in very strange conditions, that are in some way similar to first order conditions. Note that these conditions do occur if the out-of-edge layers are not in line. That happens when we combine a rectangle with a parallelogram, for example. Continuity is still possible, but these conditions become more complex as we need to take into account all affected coefficients, and correct for the difference in degree.

TD: Continuity (3)

Continuity conditions in mixed tessellations.

- Relevance of 0 in $\nu = (2, 0)$;
- Multiplex B-net shifts to allow for $r > 0$;
- Conditions are miniature simplices;
- Simplify to miniature layers.



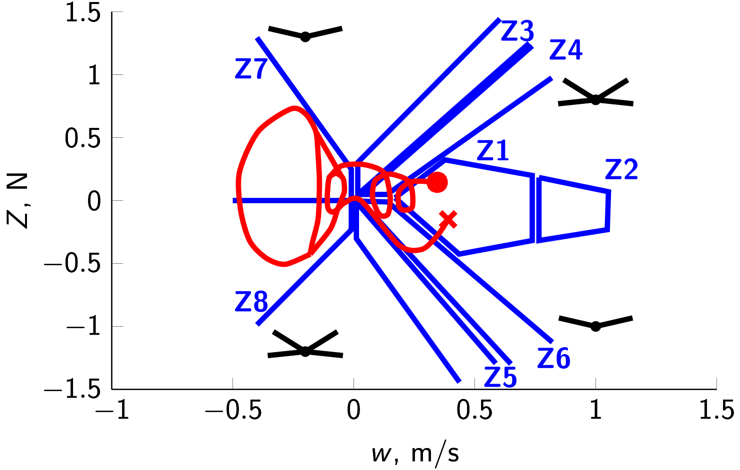
- First note that the 2-simplex is also a slice of the 3-simplex. This can be illustrated by naming it the (2,0)-multiplex. This implies we need a two-dimensional (triangle) and a zero-dimensional layer set (point) in a 3-simplex to find the 2-simplex. Note again that the zero-dimensional layer does otherwise not change anything to the definition of the polynomials on the triangle. After all, there is only one barycentric coordinate, which is constant over the entire triangle. Apart from the scaling introduced before for the (1,1)-multiplex polynomials, there is no need for any adjustments.
- Then observe that these multiplices share an edge. This is not surprising, but now we can also define it mathematically by stating that the corrections (1,1)-(0,1) and (2,0)-(1,0) both result in a (1,0)-multiplex (or 1-simplex), which is a line segment. This is indeed the shared edge.
- If we draw the continuity conditions in their most general form, we see the two miniature simplices in the B-net. By choosing the derivative direction as indicated, we can simplify these conditions greatly to only the indicated coefficients in the desired slices of the B-net.
- Note however that the degrees in the two simplices are now no longer the same! This means that for zeroth order continuity we need extra conditions to make sure the polynomial on the shared edge is also a linear one on the side of the triangle. On top of that we need to write those conditions in terms of the B-coefficients that we are actually using. This requires operations called degree reduction and degree raising respectively.
- The procedure suggested in this slide generalizes to any polynomial degree, any continuity order and any pair of multiplices that can share an edge in any number of dimensions.

Future research

- System identification;
- Nonlinear Dynamic Inversion;
- Solving PDEs;
- Mixed tessellations;
- Simplex cut splines.

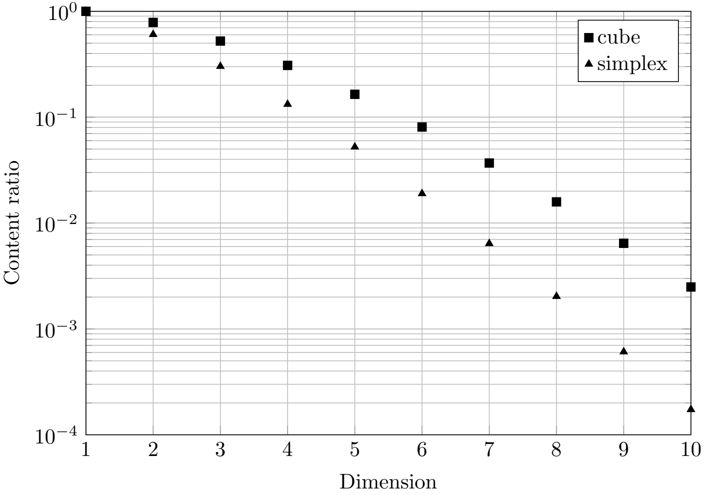
- As these splines were first applied in an engineering environment only a few months ago, there is still a lot to investigate. The first application was in system identification, which will also be the focus of my future work.
- Other application may include Nonlinear Dynamic Inversion, where a different polynomial degree to model states on the one hand and inputs on the other can be employed. Also the use in solving partial differential equations shows great promise.
- A(n even) more theoretical project is the generalization of the multiplex spline to the *simplex cut spline*. In this spline a tensor-product polynomial is defined on an arbitrary slice of a simplex. These include for example convex quadrilaterals.
- We at C&S are always looking for enthusiastic, talented MSc students willing to spend 9 months of their lives studying multivariate splines and their applications.

Extra: DelFly data set



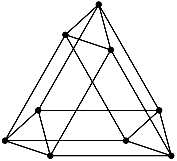
- The flight data of the DelFly has been studied and used to generate a multiplex spline model. Here you see the data points projected onto the (Z-force,w)-plane.
- From the animation it is clear that the datapoints are concentrated in different regions that can be immediately coupled to certain stages of the maneuver performed. This makes it extremely hard to construct a tessellation in which the elements are properly filled with data. The points lie in separate clouds in the high-dimensional state-space, connected by lines of data. This is horrible to fit. Ideal is a rectangular or spherical data set, which should be the prime objective during flight tests.

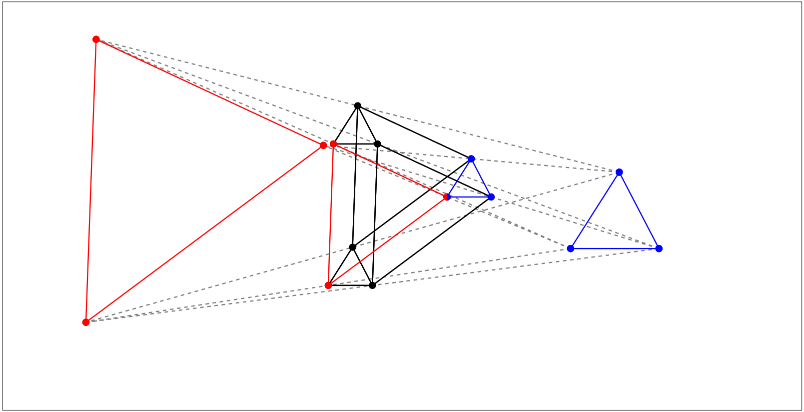
Extra: Content distribution



- It can be shown that it is easier to fill a hypercube with data than a simplex of the same dimension. One way to show this is by comparing the content distribution. That is, how is the content distributed over the geometry? Does it mainly concentrate in the middle, or does it spill out towards the vertices? If the content is concentrated in the middle, it is closer to a sphere, which is easy to fill with data.
- To quantify this, we compare the ratio of the content of the inscribed sphere to that of the geometry itself, for the simplex and the hypercube. If the ratio is close to one, the insphere is almost as big (in terms of content) as the geometry, and therefore the geometry is close to spherical.
- Note how the content ratio quickly drops to extremely low values for the n-simplex, whereas the n-cube holds out longer. At the same time it is easy to see that filling any high-dimensional multiplex with data is very difficult.

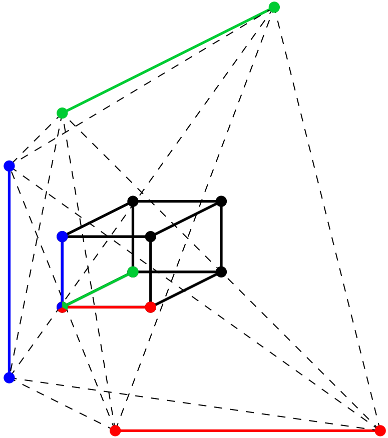
Extra: (2,2)-Multiplex





- The beauty of the multiplex spline framework is that it immediately generalizes to higher dimensions. Here we see for example a four-dimensional multiplex (note: $|\nu| = n$). It can be defined using both the bottom-up and the top-down approach.
- We can identify the two layers of the multiplex in the figure by their color. They correspond to the colors of the simplex layer sets.
- Note how the multiplex changes when we rotate and scale the layer sets.

Extra: (1,1,1)-Multiplex



- The beauty of the multiplex spline framework is that it immediately generalizes to higher dimensions. Here we see for example a three-dimensional multiplex (note: $|\nu| = n$). It can be defined using both the bottom-up and the top-down approach.
- We can identify the three layers of the multiplex in the figure (colored), and the corresponding layer sets with the same color. Note that now the vertices of the multiplex do not lie halfway the edges, but at the barycenter of a triangle connecting the corresponding vertices in the layer sets.