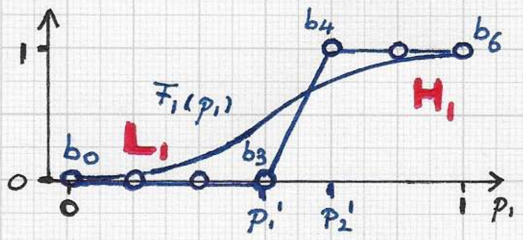
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Using Bernstein-Bézier polynomials to represent decider functions:



Sextic $F_1(p_1)$ polynomial defined as a "special sum" of Bernstein-Bézier polynomials, i.e., the sum uses only coefficients of value 0 (b_0, b_1, b_2, b_3) and value 1 (b_4, b_5, b_6). Thus, the polynomial is

$$\sum_{i=0}^{6} b_i B_i^6(p_1) = \sum_{i=4}^{6} B_i^6(p_1)$$

$$= 15(1-p_1)^2 p_1^4 + 6(1-p_1)p_1^5 + p_1^6 .$$

The graph of this polynomial has a "RAMP" region that is wide and is not steep. This shortcoming can be addressed by extending the representation to a rational Bernstein-Bézier function. By incorporating high values for weights $w_1, \ldots, w_5$ the function F_1 can be pulled to the control polygon. (Weights w_0 and w_6 have the value 1.)

The values of F_1 serve as quasi-Boolean values to be combined with values of a second function F_2.

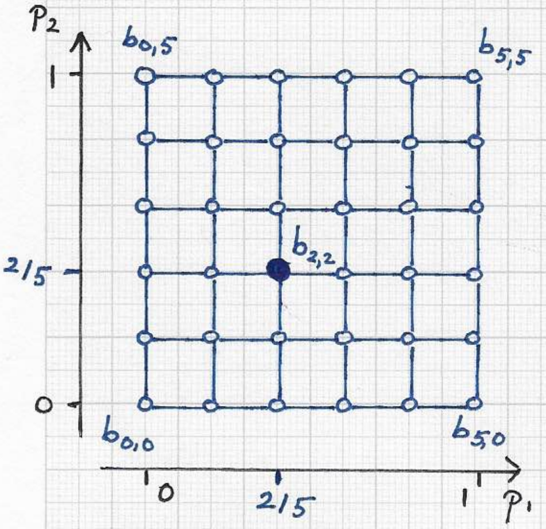
High-degree univariate Bernstein-Bézier polynomials can therefore be used to define decider functions $F_{sc}(p_{sc})$ – instead of SIGMOID functions. They provide flexibility to define the needed **L-R-H** (Low-ramp-high) functions. Nevertheless, the polynomial degree (and thus the number of control coefficients b_i) should be kept minimally, for computational efficiency. When using two univariate decider functions $F_1(p_1) = \sum_{i=0}^{n} b_i^1 B_i^n(p_1)$ and $F_2(p_2) = \sum_{i=0}^{n} b_i^2 B_i^n(p_2)$, one can use the tensor product representation

$$F(p_1, p_2) = \sum_{j=0}^{n} \sum_{i=0}^{n} b_{i,j} B_i^n(p_1) B_j^n(p_2)$$

to define the overall bivariate decider function. We must recall that $F_1(p_1)$ and $F_2(p_2)$ effectively serve the purpose of computing a (fuzzy) Boolean value, 0 (FALSE) or 1 (TRUE). Further, when combining the semantically different low and high values of F_1 and F_2 to define F, one must employ the "proper weights" such that F computes the intended Boolean output. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...



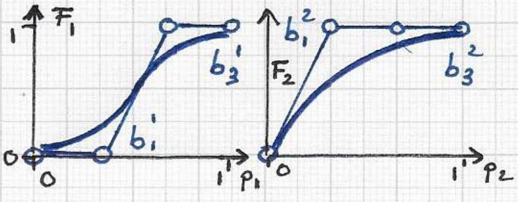
"Control net" of a biquintic, tensor product decider function

$$F(p_1,p_2)=\sum_{j=0}^{5}\sum_{i=0}^{5} b_{i,j}\, B_i^5(p_1)\, B_j^5(p_2)$$

The actual control POINTS have coordinate tuples

$$\mathbf{b}_{i,j}=\left(i/5,\ j/5,\ b_{i,j}\right)^T,$$
$$i,j=0\ldots5.$$

Additive (averaging) and multiplicative combination of two univariate decider functions:



i) $F_{add}(p_1,p_2)=\left(F_1(p_1)+F_2(p_2)\right)/2$

ii) $F_{mult}(p_1,p_2)=F_1(p_1)\cdot F_2(p_2)$

⇒ Must compute $\{b_{i,j}\}$.

The figure (left) illustrates the control coefficient/point net of a bivariate decider function in Bernstein-Bézier representation. Considering the sketched example, one must determine near-optimal values for the $6\times6=36$ control coefficients $b_{i,j}$, $i,j=0\ldots5$, where it is assumed that six coefficients b_i^1, $i=0\ldots5$, and six coefficients b_j^2, $j=0\ldots5$, define the two given univariate decider functions $F_1(p_1)$ and $F_2(p_2)$, respectively. It is interesting and important to note that the isoparametric univariate functions/curves of a bivariate function of Bernstein-Bézier form can be written as a univariate polynomial in Bernstein-Bézier form:

$$F(p_1,p_2)=\sum_{j=0}^{n}\underbrace{\sum_{i=0}^{n} b_{i,j}\, B_i^n(p_1)}\, B_j^n(p_2)$$
$$=\sum_{j=0}^{n}\ \overline{b}_j(p_1)\ B_j^n(p_2)$$

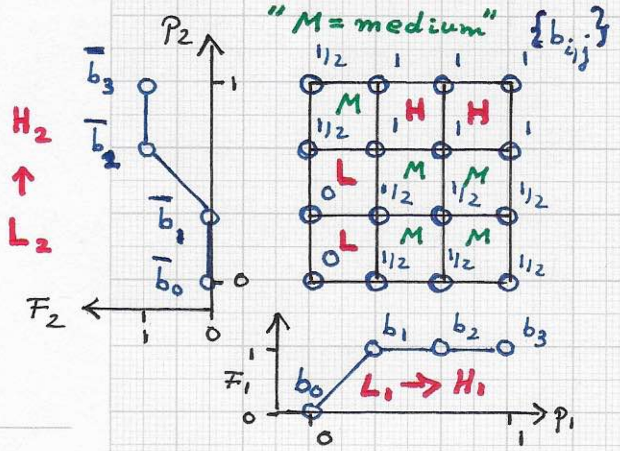
$$F(p_1,p_2)=\sum_{i=0}^{n}\underbrace{\sum_{j=0}^{n} b_{i,j}\, B_j^n(p_2)}\, B_i^n(p_1)$$
$$=\sum_{i=0}^{n}\ \overline{b}_i(p_2)\ B_i^n(p_1)$$

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Additive-averaging combination of $F_1(p_1)$ and $F_2(p_2)$ in Bernstein-Bézier form:



The two decider functions are

$$F_1(p_1) = \sum_{i=0}^{3} b_i\, B_i^3(p_1),$$
$$F_2(p_2) = \sum_{j=0}^{3} \bar{b}_j\, B_j^3(p_2).$$

i) Extruding $F_1(p_1)$ in p_2-direction replicates the b_i-values:

$$b^1_{i,j} = b_i\,, \text{ all } j.$$

Extruding $F_2(p_2)$ in p_1-direction replicates the $\bar{b}_j$-values:

$$\bar{b}^2_{I,j} = \bar{b}_j\,, \text{ all } I.$$

ii) The sum of two bivariate polynomial functions is

$$F_{sum}(p_1,p_2) = F(p_1,p_2) + \bar{F}(p_1,p_2)$$
$$= \sum_{j=0}^{n} \sum_{i=0}^{n} \left(b^1_{i,j} + \bar{b}^2_{i,j}\right) \cdot B_i^n(p_1)\, B_j^n(p_2).$$

In the context of implementing continuous, fuzzy realizations of Boolean functions, the additive-averaging and multiplicative combination of these $L_1 \rightarrow H_1$ and $L_2 \rightarrow H_2$ univariate decider functions $F_1(p_1)$ and $F_2(p_2)$, respectively, can be viewed as a ("poor") approximation of the logical **OR (v)** and the logical **AND (∧)** function. Thus, one must calculate the values of $b_{i,j}$, $i,j = 0 \ldots n$, from b_i, $i = 0 \ldots n$, and b_j, $j = 0 \ldots 5$, of the given univariate polynomials. The figure (left) illustrates additive-averaging combination of decider functions $F_1(p_1)$ and $F_2(p_2)$: **F_1 and F_2 have coefficients $\{b_i\}_{i=0}^n$ and $\{\bar{b}_j\}_{j=0}^n$. The coefficients of the combination are $\{b_{i,j}\}_{i,j=0}^n$, where**

$$b_{i,j} = (b_i + \bar{b}_j)/2\,,\quad i,j = 0 \ldots n.$$

!

Of course, it is not necessary to compute or store the $(n+1)\cdot(n+1)$ $b_{i,j}$-values; only the $(n+1)$ b_i-values and $(n+1)$ $\bar{b}_j$-values are needed. ...

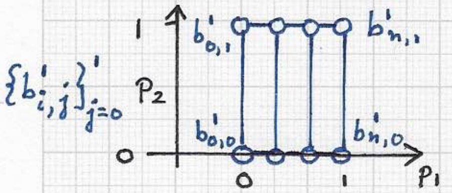
■ OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Construction of 2D control "coefficient net" from two univariate Bernstein-Bézier polynomial decider functions:

i) Given: $F_1(p_1)$

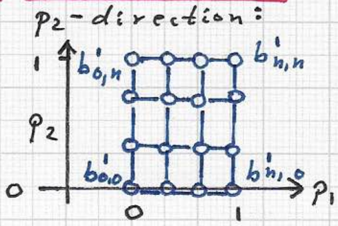
ii) "Ruled function" in p_2-direction:



$$F_1(p_1,p_2) = \sum_{j=0}^{1}\sum_{i=0}^{n} b_{i,j}^1\, B_i^n(p_1) \cdot B_j^1(p_2),$$

where $b_{i,j} = b_i$, $j=0,1$.

iii) "Knot insertion" in p_2-direction:



$$\overline{F}_1(p_1,p_2) = \sum_{j=0}^{n}\sum_{i=0}^{n} b_{i,j}^1\, B_i^n(p_1) \cdot B_j^n(p_2),$$

where $b_{i,j}^1 = b_i$, $j=0\ldots n$.

iv) Perform same steps for $F_2(p_2)$, with original coefficients b_j^2, $j=0\ldots n$:

$$\overline{F}_2(p_1,p_2) = \sum_{j=0}^{n}\sum_{i=0}^{n} b_{i,j}^2\, B_i^n(p_1) \cdot B_j^n(p_2),$$

where $b_{i,j}^2 = b_j^2$, $i=0\ldots n$.

• **Note.** While the definition of **the** additive-averaging bivariate decider function is intuitively understood, it is important to understand the actual steps involved in the construction of $F(p_1,p_2)$ from $F_1(p_1) = \sum_{i=0}^n b_i^1 B_i^n(p_1)$ and $F_2(p_2) = \sum_{j=0}^n b_j^2 B_j^n(p_2)$. Via the standard operations "extruding/ruling" and "knot insertion" one uses the given two control coefficient sets $\{b_i^1\}_{i=0}^n$ and $\{b_j^2\}_{j=0}^n$ to define two intermediate bivariate functions, i.e.,

$$\overline{F}_1(p_1,p_2) = \sum_{j=0}^n \sum_{i=0}^n b_{i,j}^1\, B_i^n(p_1)\, B_j^n(p_2),$$

with $b_{i,j}^1 = b_i$, $j=0\ldots n$, and

$$\overline{F}_2(p_1,p_2) = \sum_{j=0}^n \sum_{i=0}^n b_{i,j}^2\, B_i^n(p_1)\, B_j^n(p_2),$$

with $b_{i,j}^2 = b_j^2$, $i=0\ldots n$.

Thus, the final function $F(p_1,p_2)$ is obtained by addition and averaging applied to $\overline{F}_1$ and $\overline{F}_2$:

$$F(p_1,p_2) = \sum_{j=0}^{n}\sum_{i=0}^{n} (b_{i,j}^1 + b_{i,j}^2)/2 \cdot B_i^n(p_1)\, B_j^n(p_2).$$

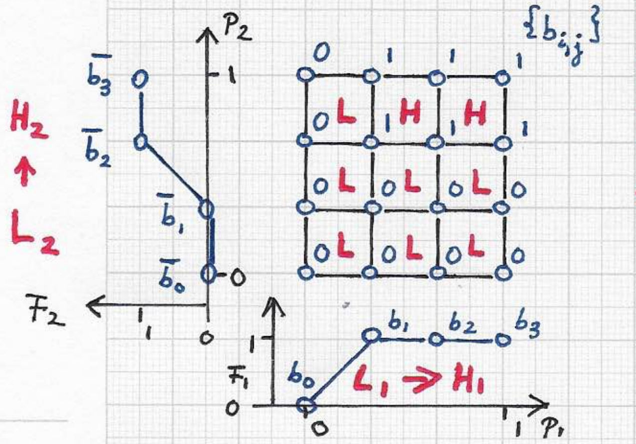
The figures (left) illustrate the steps.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Multiplicative combination of $F_1(p_1)$ and $F_2(p_2)$ in Bernstein-Bézier form:



Regions in the $[0,1]^2$ domain with relatively low and high values are indicated via L, H.

The values of coefficients $b_{i,j}$ are the products of the values of b_i and $\bar{b}_j$, $i,j = 0 \ldots n$.

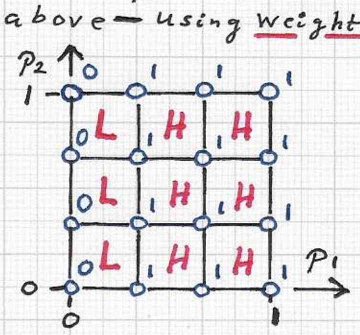
The figure (Left) sketches multiplicative combination of decider functions $F_1(p_1)$ and $F_2(p_2)$: F_1 and F_2 have coefficients $\{b_i\}_{i=0}^{n}$ and $\{\bar{b}_j\}_{j=0}^{n}$. The bivariate decider function is $F(p_1, p_2) =$

$$= F_1 \cdot F_2 = \left(\sum_{i=0}^{n} b_i B_i^n(p_1)\right)\left(\sum_{j=0}^{n} \bar{b}_j B_j^n(p_2)\right)$$

$$= \sum_{j=0}^{n} \sum_{i=0}^{n} b_i \cdot \bar{b}_j \, B_i^n(p_1) \, B_j^n(p_2)$$

$$= \sum_{j=0}^{n} \sum_{i=0}^{n} b_{i,j} \, B_i^n(p_1) \, B_j^n(p_2)$$

Special (additive) convex combination of $F_1(p_1)$ and $F_2(p_2)$ — from example above — using weight 1 and 0:



$F(p_1, p_2) = 1 \cdot F_1(p_1) + 0 \cdot F_2(p_2)$

The figures for additive-averaging and multiplicative combination of univariate decider functions demonstrate that the Boolean functions '$\vee$' and '$\wedge$' are approximated.

The example sketched in the second figure (left, bottom) shows the result of the special (additive) convex combination $F(p_1, p_2) = 1 \cdot F_1(p_1) + 0 \cdot F_2(p_2)$. One can view this combination as a numerical implementation of the logic function that yields TRUE when combining H_1 with H_2 or H_1 with L_2 and yields FALSE otherwise.

...