Fast determination of the tensorial and simplicial Bernstein enclosures

September 27, 2016

Jihad Titi Jürgen Garloff

University of Konstanz
Department of Mathematics and Statistics
and
University of Applied Sciences / HTWG Konstanz
Institute of Applied Research

Outline

- The tensorial Bernstein form of a multivariate polynomial
  - ◇ Properties
  - ◇ Computation of the tensorial Bernstein coefficients over the unit and a general box
    - ⋆ Existing methods
    - ⋆ New methods
  - ◇ Computation of the Bernstein coefficients over sub-boxes generated by subdivision
- The simplicial Bernstein form of a multivariate polynomial
- Related work

Notation

- Let n be the number of variables.
- A multi-index $(i_1, \ldots, i_n) \in \mathbb{N}^n$ is abbreviated by i. In particular, we write 0 for $(0, \ldots, 0)$.
- Comparison between and arithmetic operations with multi-indices are defined entry-wise. For $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, its *monomials* are defined as $x^i := \prod_{s=1}^{n} x_s^{i_s}$.
- For $d = (d_1, \ldots, d_n) \in \mathbb{N}^n$, we use the compact notations
  - ◇ $\sum_{i=0}^{d} := \sum_{i_1=0}^{d_1} \cdots \sum_{i_n=0}^{d_n}$
  - ◇ $\binom{d}{i} := \prod_{s=1}^{n} \binom{d_s}{i_s}$.

Tensorial Bernstein form

- Let p be an n-variate polynomial of degree l at most

$$p(x) = \sum_{i=0}^{l} a_i x^i.$$

- The i-th Bernstein polynomial of degree d, $d \geq l$, over $\boldsymbol{u} = [0, 1]^n$ is the polynomial $(0 \leq i \leq d)$

$$B_i^{(d)}(x) = \binom{d}{i} x^i (1-x)^{d-i}.$$

- The Bernstein polynomials of degree d (over $\boldsymbol{u}$) form a basis of the vector space of the polynomials of degree at most d. Therefore, p can represented by

$$p(x) = \sum_{i=0}^{d} b_i^{(d)} B_i^{(d)}(x), \quad d \geq l.$$

- The coefficients of this expansion are given by $(a_j := 0$ for $j \geq l$ and $j \neq l)$

$$b_i^{(d)} = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{d}{j}} a_j, \quad 0 \leq i \leq d.$$

(*Bernstein coefficients*).

- The Bernstein coefficients can be arranged in a multidimensional array $B(\boldsymbol{u}) = (b_i^{(d)})_{0 \leq i \leq d}$, the so-called *Bernstein patch*.

Properties of the Bernstein coefficients

- **Endpoint interpolation property (vertex values):** If $i_s \in \{0, l_s\}, s = 1, \ldots, n$, then $b_i = p(v)$, where

$$v_s = \begin{cases} 0, & i_s = 0, \\ 1, & i_s = l_s, \end{cases} \quad \text{for all } s = 1, \ldots, n.$$

- **Linearity :** Let $p = \alpha p_1 + \beta p_2,\ \alpha, \beta \in \mathbb{R}$, where l is the maximum degree of p_1 and p_2. Then

$$b_i^{(d)}(p) \ = \ \alpha b_i^{(d)}(p_1) + \beta b_i^{(d)}(p_2), \text{ for all } \ 0 \leq i \leq d,$$

where $b_i^{(d)}(p_1)$ and $b_i^{(d)}(p_2)$ are the i-th coefficients of the degree d of the Bernstein expansions of p_1 and p_2, respectively, $d \geq l$.

- The Bernstein polynomials of degree d (over $\boldsymbol{u}$) form a basis of the vector space of the polynomials of degree at most d. Therefore, p can represented by

$$p(x) = \sum_{i=0}^{d} b_i^{(d)} B_i^{(d)}(x), \;\; d \geq l.$$

- The coefficients of this expansion are given by ($a_j := 0$ for $j \geq l$ and $j \neq l$)

$$b_i^{(d)} = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{d}{j}} a_j, \;\; 0 \leq i \leq d.$$

(*Bernstein coefficients*).

- The Bernstein coefficients can be arranged in a multidimensional array $B(\boldsymbol{u}) = (b_i^{(d)})_{0 \leq i \leq d}$, the so-called *Bernstein patch*.

- **Convex hull property:** The graph of p over $\mathbf{u}$ is contained in the convex hull of the control points :

$$\left\{ \begin{pmatrix} x \\ p(x) \end{pmatrix} \ : \ x \in \boldsymbol{u} \right\} \subseteq \ \text{conv} \left\{ \begin{pmatrix} i/d \\ b_i \end{pmatrix} \ : \ 0 \leq i \leq d \right\}.$$

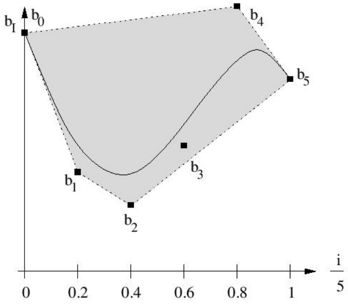


Figure: The graph of a degree 5 polynomial and the convex hull (shaded) of its control points (marked by squares).

- **Range enclosing property :** For all $x \in \boldsymbol{u}$

$$\min_{i=0}^{d} b_i^{(d)} \leq p(x) \leq \max_{i=0}^{d} b_i^{(d)}. \tag{1}$$

Equality holds in the left or right inequality in (1) if and only if the minimum or the maximum, respectively, is attained at a vertex of $\boldsymbol{u}$. This property is called **the sharpness (vertex) property**.

- **Inclusion isotonicity :** If the unit box $\boldsymbol{u}$ shrinks then the Bernstein enclosure shrinks, too.

- **Degree elevation:** The Bernstein coefficients $b_i^{(d+1)}$ of p over $\boldsymbol{u}$ can be obtained from the Bernstein coefficients $b_i^{(d)}$ as follows

$$b_i^{(d+1)} = \frac{i b_{i_{s,-1}}^{(d)} + (d+1-i) b_i^{(d)}}{d+1}, i = 0, \ldots, d+1,$$

with $b_{-1}^{(d)} = b_{d+1}^{(d)} = 0$.

- **Tensorial subdivision:** Subdivision of $\boldsymbol{u}$ in the s-th coordinate direction, $s \in \{1, \ldots, n\}$, produces two subboxes $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$,

$$\begin{aligned}\boldsymbol{u}_1 &:= [0,1] \times \ldots \times [0,\lambda] \times \ldots \times [0,1],\\ \boldsymbol{u}_2 &:= [0,1] \times \ldots \times [\lambda,1] \times \ldots \times [0,1],\end{aligned}$$

  for some $\lambda \in (0,1)$.
  The Bernstein coefficients over $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, denoted by $b_i(\boldsymbol{u}_1)$ and $b_i(\boldsymbol{u}_2)$, respectively, can be computed from the Bernstein coefficients b_i over $\boldsymbol{u}$ by using the de Casteljau algorithm.
- Subdivision is more efficient than degree elevation since iteratively applied subdivision generates a sequence of enclosures which converges quadratically to the range of p over $\boldsymbol{u}$, in contrast to linear convergence when degree elevation is applied.

Computation of the tensorial Bernstein coefficients over the unit and a general box

- Existing methods
  - ⋆ difference table method (G., 1986)
  - ⋆ Ray and Nataraj's method (matrical method 2012)
- New methods

Difference table method

- The Bernstein coefficients of a bivariate polynomial p of degree (l_1, l_2) over $\boldsymbol{u}$ are related to the coefficients of p by a forward difference operator Δ

$$\Delta_{j_1,j_2} b_0 = \frac{1}{\binom{l_1}{j_1}\binom{l_2}{j_2}} a_{j_1,j_2}.$$

- The Bernstein coefficients can be computed by the following recurrence relations :

$$\begin{aligned}
\Delta_{00} b_{i_1,i_2} &= b_{i_1,i_2}, \\
\Delta_{j_1+1,j_2} b_{i_1,i_2} &= \Delta_{j_1,j_2} b_{i_1,i_2} - \Delta_{j_1,j_2} b_{i_1+1,i_2}, \\
\Delta_{j_1,j_2+1} b_{i_1,i_2} &= \Delta_{j_1,j_2} b_{i_1,i_2} - \Delta_{j_1,j_2} b_{i_1,i_2+1},
\end{aligned}$$

where $0 \leq i_1 + 1 \leq l_1$, $0 \leq i_2 + 1 \leq l_2$, and $\Delta_{j_1,j_2} b_{0,0} := 0$ if $j_1 > l_1$ or $j_2 > l_2$.

Ray and Nataraj's method

- Ray and Nataraj proposed a matrix method for computing the Bernstein coefficients of a multivariate polynomial over the unit and a general box.
- Their method involves only matrix operations such as multiplication, inversion, transposition, and reshaping.
- The method is obtained by equating the matrix representations of a polynomial in power form and in Bernstein form and solving for the Bernstein coefficients.

New methods

We propose a method for the computation of the Bernstein coefficients of a given multivariate polynomial over the unit box and a general box which is superior to the two former methods.

Matrix methods over the unit box

- Bivariate case
  - ◇ Matrical description of difference table method
    We start by arranging the coefficients of p in a matrix A as follows :

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,l_2} \\ a_{1,0} & a_{1,1} & \dots & a_{1,l_2} \\ \vdots & \vdots & \dots & \vdots \\ a_{l_1,0} & a_{l_1,1} & \dots & a_{l_1,l_2} \end{bmatrix}.$$

⋄ Let the entries of the matrix $\Lambda(\boldsymbol{u}) = (\lambda_{j_1,j_2})$ be given by

$$\lambda_{j_1,j_2} := \frac{a_{j_1-1,j_2-1}}{\binom{l_1}{j_1-1}\binom{l_2}{j_2-1}}, \text{where } j_s = 1, \ldots, l_s + 1,\ s = 1, 2.$$

⋄ Starting from row $l_s - \mu + 2$, addition of each row to the following one, $\mu = 1, \ldots, l_s$, is described by left multiplication by the matrices

$$K_\mu^s := \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow l_s - \mu + 2 \\ \\ \\ \\ \end{matrix}$$

Then the Bernstein patch is given as

$$B(\boldsymbol{u}) = (K_1^2 \cdots K_{l_2}^2 (K_1^1 \cdots K_{l_1}^1 \Lambda(\boldsymbol{u}))^T)^T.$$

For the *lower triangular Pascal matrix*

$$P_s := \begin{bmatrix} \binom{0}{0} & 0 & \dots & 0 \\ \binom{1}{0} & \binom{1}{1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \binom{l_s}{0} & \binom{l_s}{1} & \dots & \binom{l_s}{l_s} \end{bmatrix}$$

the factorization

$$P_s = \prod_{\mu=1}^{l_s} K_\mu^s$$

holds. Therefore,

$$B(\boldsymbol{u}) = (P_2(P_1\Lambda(\boldsymbol{u}))^T)^T.$$

- Multivariate case

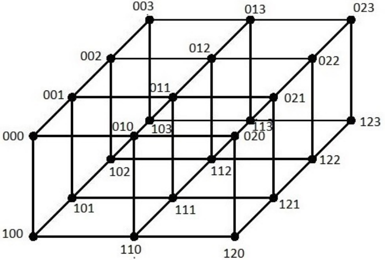
◇ Let p be an n-variate polynomial of degree $l = (l_1, \ldots, l_n)$.

◇ The coefficients of p are arranged in an $(l_1 + 1) \times l^*$ matrix, where $l^* := \prod_{i=2}^{n}(l_i + 1)$, as follows

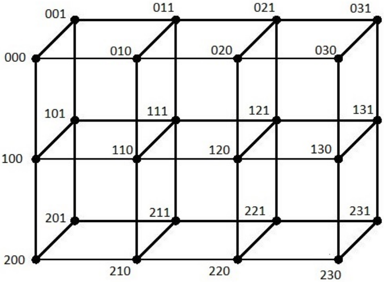
$$\begin{bmatrix} a_{0,0,0,\ldots,0} & a_{0,1,0,\ldots,0} & \cdots & a_{0,l_2,0,\ldots,0} & \cdots & a_{0,0,l_3,\ldots,0} & a_{0,1,l_3,\ldots,0} & \cdots & a_{0,l_2,l_3,\ldots,l_n} \\ a_{1,0,0,\ldots,0} & a_{1,1,0,\ldots,0} & \cdots & a_{1,l_2,0,\ldots,0} & \cdots & a_{1,0,l_3,\ldots,0} & a_{1,1,l_3,\ldots,0} & \cdots & a_{1,l_2,l_3,\ldots,l_n} \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots & \vdots & \ldots & \vdots \\ a_{l_1,0,0,\ldots,0} & a_{l_1,1,0,\ldots,0} & \cdots & a_{l_1,l_2,0,\ldots,0} & \cdots & a_{l_1,0,l_3,\ldots,0} & a_{l_1,1,l_3,\ldots,0} & \cdots & a_{l_1,l_2,l_3,\ldots,l_n} \end{bmatrix}$$

◇ The superscript c denotes the *cyclic ordering* of the sequence of the indices.

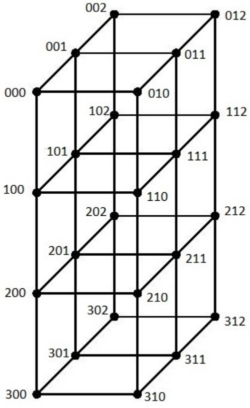
Cyclic ordering for a three-dimensional array with $l_1 = 1, l_2 = 2, l_3 = 3$



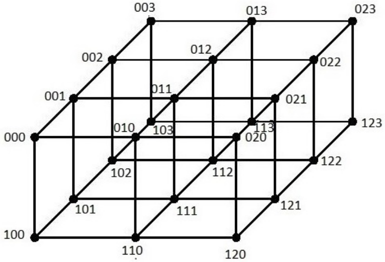
(a) $\Lambda(\boldsymbol{u})$



(b) $(P_1\Lambda(\boldsymbol{u}))^c$



(c) $(P_2(P_1\Lambda(\boldsymbol{u}))^c)^c$



(d) $(P_3(P_2(P_1\Lambda(\boldsymbol{u}))^c)^c)^c$ providing $B(\boldsymbol{u})$

- The matrix $\Lambda(\boldsymbol{u})$ is obtained from A by multiplying $a_{i_1,\ldots,i_n}$ by $\binom{l_1}{i_1}^{-1} \cdots \binom{l_n}{i_n}^{-1}$.
- We put

$$\begin{aligned} \Lambda_0 &:= \Lambda(\boldsymbol{u}), \\ \Lambda_s &:= (P_s \Lambda_{s-1}(\boldsymbol{u}))^c,\ s = 1, \ldots, n. \end{aligned}$$

- The Bernstein patch $B(\boldsymbol{u})$ arranged accordingly in an $(l_1 + 1) \times l^*$ matrix is given by Λ_n.

Matrix methods for a general box

⋆ We affinely map a given box $\boldsymbol{x} = [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_n, \overline{x}_n]$ to $\boldsymbol{u}$ by

$$z_s = \frac{x_s - \underline{x}_s}{\overline{x}_s - \underline{x}_s}, \quad s = 1, \ldots, n. \tag{2}$$

⋆ With $D_s(t) := \operatorname{diag}(1, t, t^2, \ldots, t^\kappa)$ put

$$Q_s := \begin{cases} D_s(\frac{\overline{x}_s - \underline{x}_s}{\underline{x}_s}) P_s^T D_s(\underline{x}_s), & \underline{x}_s \neq 0, \\ D_s(\overline{x}_s), & \underline{x}_s = 0. \end{cases}$$

⋆ After substituting (2) in p we obtain a polynomial p^* over $\boldsymbol{u}$. The coefficients of p^* are arranged in a matrix, A^* say, which is given by

$$A^* = (Q_n(\ldots(Q_2(Q_1 A)^c)^c \ldots)^c)^c.$$

⋆ Then apply the procedure for the unit box to A^*.

Amount of arithmetic operations

Unit box

- For simplicity and ease of comparison between the existing methods, we assume that $l_1 = \cdots = l_n = \kappa$.
- Some basic quantities like factorials and binomial coefficients are precomputed.
- We present two new methods together with their complexity by which we can carry out the matrix multiplication in

$$\Lambda_n = (P_n(\cdots(P_2(P_1\Lambda(\boldsymbol{u}))^c)^c\cdots)^c)^c.$$

Method 1

- We use the factorization

$$P = K_1 K_2 \cdots K_\kappa.$$

  This factorization allows us to get rid of the multiplication operations.
- Method 1 is superior to method (R.N) if $\kappa > 1$ (the complexity of both methods is $O(n\kappa^{n+1})$).

TABLE – Comparison between the number of arithmetic operations of the three methods

Method	number of additions	number of multiplications/divisions
R.N	$n\frac{\kappa(\kappa+1)^n}{2}$	$\frac{\kappa(\kappa-1)}{2} + n\frac{\kappa(\kappa+1)^n}{2}$
Method 1 and difference table method	$n\frac{\kappa(\kappa+1)^n}{2}$	$n(\kappa+1)^n$

Method 2

- We use the factorization

$$P = GTG^{-1},$$

where $G := \mathrm{diag}(1, 1, 2!, \ldots, \kappa!)$ and T is the lower triangular Toeplitz matrix

$$T := \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 1 & 0 & \ldots & 0 & 0 \\ \frac{1}{2!} & 1 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{(\kappa-1)!} & \frac{1}{(\kappa-2)!} & \frac{1}{(\kappa-3)!} & \ldots & 1 & 0 \\ \frac{1}{\kappa!} & \frac{1}{(\kappa-1)!} & \frac{1}{(\kappa-2)!} & \ldots & 1 & 1 \end{bmatrix}.$$

- The factorials $i_\varrho!$ appearing in $\binom{\kappa}{i_\varrho}^{-1}$ and on the diagonal of G^{-1} cancel out, $\varrho = 1, \ldots, n$.
- Multiply $a_{i_1,\ldots,i_n}$ by $\frac{(\kappa-i_1)!}{\kappa!} \ldots \frac{(\kappa-i_n)!}{\kappa!}$ and name the resulting matrix $\Lambda'(\boldsymbol{u})$ then

$$(\underbrace{GT(\ldots(GT(GT}_{n\ times}\Lambda'(\boldsymbol{u}))^c)^c\ldots)^c)^c$$

gives the Bernstein patch in matrix form.
- The computation of the matrix-vector multiplication with the matrix T can be carried out by using the Fast Fourier Transform (FFT).
- Then the total number of operations required for the computation of $B(\boldsymbol{u})$ can be reduced to $O(n\kappa^n \log_2 \kappa)$, whereas the complexity of Method 1 is $O(n\kappa^{n+1})$, so Method 2 has lowest complexity.

General box

Method 1

TABLE – Number of arithmetic operations required to obtain the Bernstein patch over a general box by (Method 1) and (R.N)

Calculation of	number of additions	number of multiplications/divisions
$D(\underline{x}_s)$, $D(\frac{\overline{x}_s - \underline{x}_s}{\underline{x}_s})$	n	$2n(\kappa-1)+n$
A^*	$n\kappa\frac{(\kappa+1)^n}{2}$	$2n(\kappa+1)^n$
Method 1 over $\boldsymbol{u}$ (starting from A^*)	$n\kappa\frac{(\kappa+1)^n}{2}$	$n(\kappa+1)^n$
total number of Method 1	$n\kappa(\kappa+1)^n+n$	$3n(\kappa+1)^n+2n(\kappa-1)+n$
R.N	$n+\frac{n\kappa(\kappa+1)(2\kappa+1)}{6}+n\kappa(\kappa+1)^n$	$\frac{\kappa(\kappa-1)}{2}+\kappa n(\kappa+3)+\frac{n(\kappa+1)(\kappa+2)(2\kappa+3)}{6}+n(\kappa+1)^{n+1}$

The complexity of Method 1 and (R.N) is $O(n\kappa^{n+1})$ but Method 1 is superior for $\kappa > 1$.

Tensorial subdivision

The Bernstein coefficients over the two sub-boxes $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ resulting from subdivision of the original box $\boldsymbol{u}$ in the s-th direction, $s \in \{1, 2, \ldots, n\}$, are obtained from the de Casteljau algorithm :

1. Put $b_i^{[0]} := b_i, \; 0 \leq i \leq l$.
2. For $\nu = 1, \ldots, l_s$:

$$b_i^{[\nu]} := \left\{ \begin{array}{ll} b_i^{[\nu-1]}, & \text{if } \; i_s < \nu \\ (1-\lambda) b_{i_1,\ldots,i_{s,}-1,\ldots,i_n}^{[\nu-1]} + \lambda b_i^{[\nu-1]}, & \text{if } \; i_s \geq \nu \end{array} \right\}, \; 0 \leq i \leq l.$$

3. $b_i(\boldsymbol{u}_1) := b_i^{[l_s]}$; $b_i(\boldsymbol{u}_2) := b_{i_1,\ldots,l_s,\ldots,i_n}^{[l_s - i_s]}, \; 0 \leq i \leq l$.

The Bernstein coefficients $b_i(\boldsymbol{u}_2)$ on the sub-box $\boldsymbol{u}_2$ are obtained as intermediate values of this computation, since for $\nu = 0, \ldots, l_s$ the following relation holds $b_{i_1,\ldots,l_s-\nu,\ldots,i_n}(\boldsymbol{u}_2) = b_{i_1,\ldots,l_s,\ldots,i_n}^{[\nu]}(\boldsymbol{u})$.

Tensorial subdivision in matrix form

The Bernstein coefficients over the two sub-boxes $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are arranged in matrices $\mathcal{B}(\boldsymbol{u}_1)$ and $\mathcal{B}(\boldsymbol{u}_2)$.
From the de Casteljau algorithm, we obtain $l_s + 1$ matrices. In the ν-th step, $\mathcal{B}^{[\nu]}(\boldsymbol{u})$ is given as follows:

The first ν rows are identical with the first ν rows in the $(\nu - 1)$-th step and the remaining rows are obtained as a convex combination of two consecutive rows.

The Bernstein patch $\mathcal{B}(\boldsymbol{u}_1)$ over the sub-box $\boldsymbol{u}_1$ is provided by the last matrix $\mathcal{B}^{[l_s]}(\boldsymbol{u})$.
The i-th row, $i = 1, \ldots, l_s + 1$, of the Bernstein coefficients matrix $\mathcal{B}(\boldsymbol{u}_2)$ over the sub-box $\boldsymbol{u}_2$ is equal to the last row in $\mathcal{B}^{[l_s - i + 1]}(\boldsymbol{u})$.
Without loss of generality we subdivide $\boldsymbol{u}$ in the direction along the first coordinate direction $(s = 1)$; the subdivision in the other directions is done by firstly considering the cyclic ordering.

- In matrix language, these manipulations are described by pre-multiplication of the matrix $\mathcal{B}(\boldsymbol{u})$ by the following matrices.
- For simplicity we assume that $l_1 = \ldots = l_n = \kappa$.
- For $\mu = 1, 2, \ldots, \kappa$ we define the matrices L_μ and $R_\mu \in \mathbb{R}^{\kappa+1,\kappa+1}$

$$L_\mu := \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1-\lambda & \lambda & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1-\lambda & \lambda & \cdots & 0 \\ \vdots & \vdots & \ldots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1-\lambda & \lambda \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow l_s - \mu + 2 \\ \\ \\ \\ \end{matrix}$$

and

$$R_\mu := \begin{bmatrix} 1-\lambda & \lambda & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1-\lambda & \lambda & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \cdots & 1-\lambda & \lambda & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \leftarrow \mu + 1$$

- Then we obtain the Bernstein patches over $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ by

$$\begin{aligned} \mathcal{B}(\boldsymbol{u}_1) &= L_1 L_2 \dots L_\kappa \mathcal{B}(\boldsymbol{u}), \\ \mathcal{B}(\boldsymbol{u}_2) &= R_1 R_2 \dots R_\kappa \mathcal{B}(\boldsymbol{u}). \end{aligned}$$

- We define $L^\dagger \in \mathbb{R}^{\kappa+1,\kappa+1}$ by

$$L^\dagger := \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1-\lambda & \binom{1}{1}\lambda & 0 & \dots & 0 \\ (1-\lambda)^2 & \binom{2}{1}(1-\lambda)\lambda & \binom{2}{2}\lambda^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (1-\lambda)^\kappa & \binom{\kappa}{1}(1-\lambda)^{\kappa-1}\lambda & \binom{\kappa}{2}(1-\lambda)^{\kappa-2}\lambda^2 & \dots & \binom{\kappa}{\kappa}\lambda^\kappa \end{bmatrix}.$$

- We have the factorization

$$L^\dagger = \prod_{\mu=1}^{\kappa} L_\mu = D(1-\lambda)PD(\frac{\lambda}{1-\lambda}).$$

- Then $\mathcal{B}(\boldsymbol{u}_1)$ can be represented as

$$\mathcal{B}(\boldsymbol{u}_1) = D(1-\lambda)PD(\frac{\lambda}{1-\lambda})\mathcal{B}(\boldsymbol{u}).$$

- By using $P = GTG^{-1}$, we have

$$\mathcal{B}(\boldsymbol{u}_1) = D^{(2)} T D^{(1)} \mathcal{B}(\boldsymbol{u}),$$

where $D^{(2)} := D(1-\lambda)G$ and $D^{(1)} := G^{-1} D(\frac{\lambda}{1-\lambda})$.

- In the same way $\mathcal{B}(\boldsymbol{u}_2)$ can be obtained.

TABLE – Comparison between the de Casteljau algorithm and the proposed method for the computation of $\mathcal{B}(\boldsymbol{u}_1)$ and $\mathcal{B}(\boldsymbol{u}_2)$

Calculation of	number of additions	number of multiplications/ divisions
de Casteljau algorithm	$\kappa\frac{(\kappa+1)^n}{2}$	$\kappa(\kappa+1)^n$
$\mathcal{B}(\boldsymbol{u}_1)$	$\kappa\frac{(\kappa+1)^n}{2}$	$2\kappa(\kappa+1)^{n-1}+2\kappa$
$\mathcal{B}(\boldsymbol{u}_2)$	$\kappa\frac{(\kappa+1)^n}{2}$	$(\kappa+1)^{n-1}(2\kappa+1)+3\kappa$
$\mathcal{B}(\boldsymbol{u}_1)$ and $\mathcal{B}(\boldsymbol{u}_2)$ together	$\kappa^2(\kappa+1)^{n-1}$	$3\kappa(\kappa+1)^{n-1}+4\kappa$

J. Titi and J. Garloff, Matrix Methods for the Tensorial Bernstein Form and for the Evaluation of Multivariate Polynomials, submitted.

The simplicial Bernstein form

- Let $v_0, v_1, \ldots, v_n$ be $n+1$ affinely independent points of $\mathbb{R}^n$. Their convex hull is called *simplex of vertices* $v_0, v_1, \ldots, v_n$ and denoted by $V = [v_0, v_1, \ldots, v_n]$.
- Any vector $x \in \mathbb{R}^n$ can be written as an affine combination of the vertices $v_0, v_1, \ldots, v_n$ with weights $\lambda_0, \ldots, \lambda_n$ called *barycentric coordinates*.
- If $x = (x_1, \ldots, x_n) \in V$, then $\lambda = (\lambda_0, \ldots, \lambda_n) = (1 - \sum_{i=1}^{n} x_i, x_1, \ldots, x_n)$.
- For every multi-index $\alpha = (\alpha_0, \ldots, \alpha_n) \in \mathbb{N}^{n+1}$ we put $\hat{\alpha} := (\alpha_1, \ldots, \alpha_n)$ and we write $|\alpha| := \alpha_0 + \ldots + \alpha_n$.

The simplicial Bernstein form

- If $|\alpha| = k$, we further use the notation $\binom{k}{\alpha} := \frac{k!}{\alpha_0! \ldots \alpha_n!}$.
- We will consider here only the *standard simplex* $\Delta := [0, e_1, \ldots, e_n]$, where e_s is the s^{th} vector of the canonical basis of $\mathbb{R}^n$, $s = 1, \ldots, n$. This is no restriction since any non-degenerate simplex V in $\mathbb{R}^n$ can be mapped affinely upon Δ.

- **Results :**
  Matrix methods for the computation of the Bernstein coefficients and of the Bernstein coefficients on subsimplices generated by bisection of Δ.

- Bernstein polynomials of degree k over Δ are the polynomials $(B_\alpha^{(k)})_{|\alpha|=k}$, defined as

$$B_\alpha^{(k)} := \binom{k}{\alpha} \lambda^\alpha.$$

- The Bernstein polynomials of degree k form a basis of the vector space of polynomials of degree at most k. Therefore, p can be uniquely represented as

$$p(x) = \sum_{|\alpha|=k} b_\alpha^{(k)} B_\alpha^{(k)}.$$

- The coefficients of this expansion are given by

$$b_\alpha^{(k)} = \sum_{\hat{\beta} \le \hat{\alpha}} \frac{\binom{\hat{\alpha}}{\hat{\beta}}}{\binom{k}{\beta}} a_{\hat{\beta}},\; a_{\hat{\beta}} := 0 \text{ for } \hat{\beta} \ge l,\; \hat{\beta} \ne l.$$

(Bernstein coefficients)

Related work

- Computation (without overestimation) of the set of all tensorial Bernstein coefficients of an interval polynomial
- Matrix methods for the evaluation of multivariate polynomial in the
  - ◇ power representation (Horner scheme)
  - ◇ tensorial Bernstein representation
  - ◇ simplicial Bernstein representation

THANK YOU VERY MUCH