



Symbolic-Numeric Computation of the Bernstein Coefficients of a Polynomial from Those of One of Its Partial Derivatives and of the Product of Two Polynomials

Jihad Titi¹ and Jürgen Garloff^{2,3}(✉)

¹ Department of Applied Mathematics and Physics, Palestine Polytechnic University,
Hebron, Palestine

jihadtiti@yahoo.com

² Department of Mathematics and Statistics, University of Konstanz,
78464 Konstanz, Germany

³ University of Applied Sciences/HTWG Konstanz, Institute for Applied Research,
Alfred-Wachtel-Str. 8, 78462 Konstanz, Germany
juergen.garloff@htwg-konstanz.de

Abstract. The expansion of a given multivariate polynomial into Bernstein polynomials is considered. Matrix methods for the calculation of the Bernstein expansion of the product of two polynomials and of the Bernstein expansion of a polynomial from the expansion of one of its partial derivatives are provided which allow also a symbolic computation.

Keywords: Multivariate polynomial · Bernstein polynomial · Bernstein coefficient

MSC 2010: 65F30 · 41-04

1 Introduction

In this paper, we consider the expansion of a multivariate polynomial into Bernstein polynomials over a box, i.e., an axis-aligned region, in \mathbb{R}^n . This expansion has many applications, e.g., in computer aided geometric design, robust control, global optimization, differential and integral equations, finite element analysis [6]. A very useful property of this expansion is that the interval spanned by the minimum and maximum of the coefficients of this expansion, the so-called Bernstein coefficients, provides bounds for the range of the given polynomial over the considered box, see, e.g., [8, 10]. A simple (but by no means economic) method for the computation of the Bernstein coefficients from the coefficients of the given polynomial is the use of formula (5) below. This formula (and also similar ones for the Bernstein coefficients over more general sets like simplexes and polytopes) allows the symbolic computation of these quantities when the coefficients of the

given polynomial depend on parameters. Some applications are making use of this symbolic computation: in [4, Sections 3.2 and 3.3], [5], see also the many references therein, the reachability computation and parameter synthesis with applications in biological modelling are considered. In [2, 3], parametric polynomial inequalities over parametric boxes and polytopes are treated. Applications in static program analysis and optimization include dependence testing between references with linearized subscripts, dead code elimination of conditional statements, and estimation of memory requirements in the development of embedded systems. Applications which involve polynomials of higher degree or many variables require a computation of the Bernstein coefficients which is more economic than by formula (5). In [14] and [16], we have presented a matrix method for the computation of the Bernstein coefficients which is faster than the methods developed so far. In this paper, we consider firstly the case where we have already computed the Bernstein coefficients of two multivariate polynomials and wish to compute the Bernstein expansion of their product for which we present two approaches. For the univariate case see [7, Subsection 4.2]. Secondly, we show how the Bernstein coefficients of a multivariate polynomial can be computed from the Bernstein coefficients of its partial derivatives. This problem appears for example when bounds for the range of a complex polynomial over a rectangular region in the complex plane are wanted and the Cauchy-Riemann equations are employed, see [13, Section 4.3], [17], but it is also of interest by its own.

The organization of our paper is as follows. In the next section, we introduce the notation which is used throughout the paper. In Sect. 3, we first briefly recall the expansion of a multivariate real polynomial into Bernstein polynomials over a box and some of its fundamental properties. In the second part, we recall from [14, 16] a matrix method for the computation of the Bernstein coefficients. In Sect. 4 we present two matrix methods for the computation of the Bernstein coefficients of the product of two polynomials and in Sect. 5 the computation of the Bernstein coefficients of a polynomial from those of one of its partial derivatives.

2 Notation

In this section, we introduce the notation that we are using throughout this paper. Let $n \in \mathbb{N}$ (set of the nonnegative integers) be the number of variables. A multi-index $(i_1, \dots, i_n) \in \mathbb{N}^n$ is abbreviated by i . In particular, we write 0 for $(0, \dots, 0)$ and e^s for the multi-index that has a 1 in position s and 0's otherwise. Arithmetic operations with multi-indices are defined entry-wise; the same applies to comparison between multi-indices. For the multi-index $i = (i_1, \dots, i_s, \dots, i_n)$ we define $i_{s,q} := (i_1, \dots, i_s + q, \dots, i_n)$ and $i_{[s,q]} := (i_1, \dots, q, \dots, i_n)$, $s \in \{1, \dots, n\}$, $q \in \mathbb{Z}$. For $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, its *monomials* are defined as $x^i := \prod_{s=1}^n x_s^{i_s}$. For $d = (d_1, \dots, d_n) \in \mathbb{N}^n$ such that $i \leq d$, we use the compact notations

$$\sum_{i=0}^d := \sum_{i_1=0}^{d_1} \dots \sum_{i_n=0}^{d_n}, \binom{d}{i} := \prod_{s=1}^n \binom{d_s}{i_s}.$$

For the ease of presentation, we index all array entries starting from zero.

3 Bernstein Expansion

3.1 Bernstein Representation over the Unit Box

In this section, we present fundamental properties of the Bernstein expansion over a box, e.g., [6, Subsection 5.1], [8, 10], that are employed throughout the paper. For simplicity we consider the unit box $\mathbf{u} := [0, 1]^n$, since any compact nonempty box \mathbf{x} of \mathbb{R}^n can be mapped affinely onto \mathbf{u} . Let $\ell \in \mathbb{N}^n$, $a_j \in \mathbb{R}$, $j = 0, \dots, \ell$, such that for $s = 1, \dots, n$

$$\ell_s := \max \{q \mid a_{j_1, \dots, j_{s-1}, q, j_{s+1}, \dots, j_n} \neq 0\}. \tag{1}$$

Let p be an ℓ -th degree n -variate polynomial with the power representation

$$p(x) = \sum_{j=0}^{\ell} a_j x^j. \tag{2}$$

We expand p into Bernstein polynomials of degree d , $d \geq \ell$, over \mathbf{u} as

$$p(x) = \sum_{j=0}^d b_j^{(d)} B_j^{(d)}(x), \tag{3}$$

where $B_j^{(d)}$ is the j -th Bernstein polynomial of degree d , defined as

$$B_j^{(d)}(x) := \binom{d}{j} x^j (1-x)^{d-j}, \tag{4}$$

and $b_j^{(d)}$ is the j -th Bernstein coefficient of p of degree d over \mathbf{u} which is given by

$$b_j^{(d)} = \sum_{i=0}^j \binom{j}{i} a_i, \quad 0 \leq j \leq d, \tag{5}$$

with the convention that $a_i := 0$ if $i \geq \ell, i \neq \ell$. We call (3) the *Bernstein representation of p* and arrange the Bernstein coefficients in a multidimensional array $B(\mathbf{u}) = (b_j^{(d)})_{0 \leq j \leq d}$, the so-called *Bernstein patch*. Note that the Bernstein coefficients lying on the vertices of $B(\mathbf{u})$ are values of p at the respective vertices of \mathbf{u} . More generally, the Bernstein coefficients on an r -dimensional face of \mathbf{u} , $r = 0, 1, \dots, n-1$, are just the Bernstein coefficients lying on the respective faces of $B(\mathbf{u})$ [9, Lemma 2]. E.g., assume that \mathbf{v} is an $(n-1)$ -dimensional face of \mathbf{u} that is obtained by setting $x_s = 0$ or 1 , for some $s \in \{1, \dots, n\}$. For $i \in \mathbb{N}^n$ and $r \in \mathbb{N}$ we define

$$i_{[s,r]} := (i_1, \dots, i_{s-1}, r, i_{s+1}, \dots, i_n). \tag{6}$$

Then, the Bernstein coefficients of p over \mathbf{v} are given by

$$b_i^{(d)}(p, \mathbf{v}) = \begin{cases} b_{i_{[s,0]}} & \text{if } x_s = 0, \\ b_{i_{[s,d_s]}} & \text{if } x_s = 1. \end{cases}$$

3.2 Computation of the Bernstein Coefficients

We recall from [16] a method for the computation of the Bernstein coefficients of the n -variate polynomial p given in (2).

Matrix Method for the Unit Box. The superscript c denotes the *cyclic ordering* of the sequence of the indices, i.e., the order of the indices of the entries of the array under consideration is changed cyclically. This means that the index in the first position is replaced by the index in the second one, the index in the second position by the one in the third, \dots , the index in the n -th position by the one in the first position (see Fig. 1 as an illustration in the trivariate case). So after n cyclic orderings the sequence of the indices is again in its initial order. Note that in the bivariate case the cyclic ordering is just the usual matrix transposition.

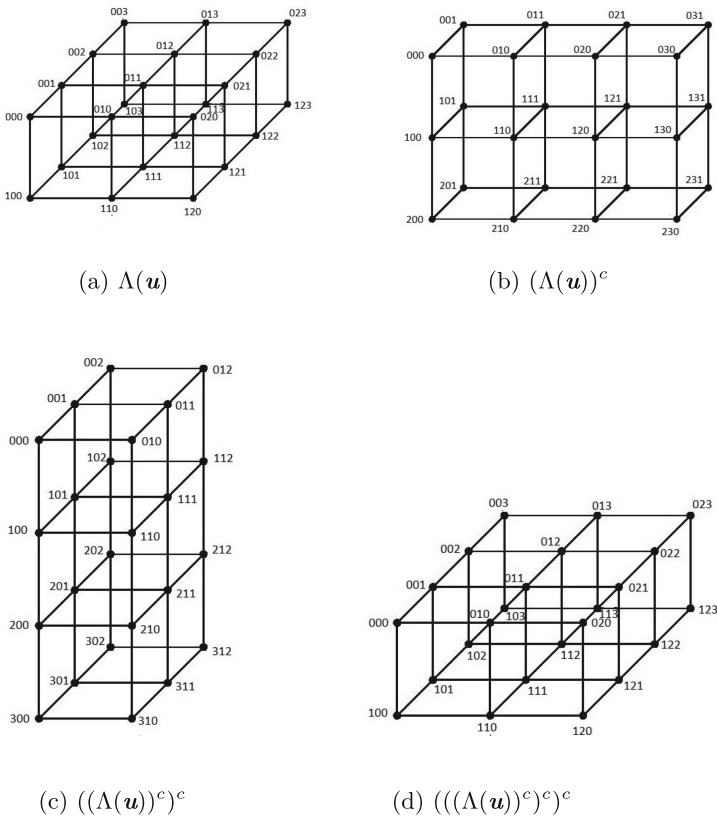


Fig. 1. Cyclic ordering of a three-dimensional array with $\ell_1 = 1$, $\ell_2 = 2$, and $\ell_3 = 3$

The coefficients of p are arranged in an $(\ell_1 + 1) \times \ell^*$ matrix A , where $\ell^* := \prod_{s=2}^n (\ell_s + 1)$. The correspondence between the coefficients a_j of p and the entry of A in row i and column j is as follows:

$$i = j_1, \tag{7a}$$

$$j = j_2 + \sum_{s=3}^n j_s (\ell_2 + 1) \cdot \dots \cdot (\ell_{s-1} + 1). \tag{7b}$$

Then A can be represented as the matrix

$$\begin{bmatrix} a_{0,0,0,\dots,0} & a_{0,1,0,\dots,0} \cdots a_{0,\ell_2,0,\dots,0} & a_{0,0,1,\dots,0} \cdots a_{0,\ell_2,1,\dots,0} \cdots \\ a_{1,0,0,\dots,0} & a_{1,1,0,\dots,0} \cdots a_{1,\ell_2,0,\dots,0} & a_{1,0,1,\dots,0} \cdots a_{1,\ell_2,1,\dots,0} \cdots \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots \\ a_{\ell_1,0,0,\dots,0} & a_{\ell_1,1,0,\dots,0} \cdots a_{\ell_1,\ell_2,0,\dots,0} & a_{\ell_1,0,1,\dots,0} \cdots a_{\ell_1,\ell_2,1,\dots,0} \cdots \\ \dots & a_{0,0,\ell_3,\dots,0} \cdots a_{0,\ell_2,\ell_3,\dots,0} \cdots a_{0,0,\ell_3,\dots,\ell_n} & a_{0,1,\ell_3,\dots,\ell_n} \cdots a_{0,\ell_2,\ell_3,\dots,\ell_n} \\ \dots & a_{1,0,\ell_3,\dots,0} \cdots a_{1,\ell_2,\ell_3,\dots,0} \cdots a_{1,0,\ell_3,\dots,\ell_n} & a_{1,1,\ell_3,\dots,\ell_n} \cdots a_{1,\ell_2,\ell_3,\dots,\ell_n} \\ \dots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & \dots \\ \dots & a_{\ell_1,0,\ell_3,\dots,0} \cdots a_{\ell_1,\ell_2,\ell_3,\dots,0} \cdots a_{\ell_1,0,\ell_3,\dots,\ell_n} & a_{\ell_1,1,\ell_3,\dots,\ell_n} \cdots a_{\ell_1,\ell_2,\ell_3,\dots,\ell_n} \end{bmatrix}. \tag{8}$$

The matrix $\Lambda(\mathbf{u})$ is obtained from A by multiplying a_j by $\binom{\ell}{j}^{-1}$. We put $\Lambda_0 := \Lambda(\mathbf{u})$ and define for $s = 1, \dots, n$

$$\Lambda_s := (P_s \Lambda_{s-1})^c, \tag{9}$$

where P_s is the lower triangular Pascal matrix,

$$(P_s)_{ij} := \begin{cases} \binom{i}{j}, & \text{if } j \leq i, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

In (9), the matrix multiplication is performed according to the factorization, e.g., [1, Lemma 1],

$$P_s = \prod_{\mu=1}^{\ell_s} K_\mu^s, \tag{11}$$

where the bidiagonal matrices K_μ^s , $\mu = 1, \dots, \ell_s$, are given by

$$(K_\mu^s)_{ij} := \begin{cases} 1, & \text{if } i = j, \\ 1, & \text{if } i = j + 1, \ell_s - \mu \leq j \leq \ell_s - 1, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Define for $s = 1, \dots, n$, $r := s \bmod n$. Then for $s = 1, \dots, n$, the entry in position (v_1, v_2) in Λ_{s-1} becomes (v'_1, v'_2) in Λ_s , where

$$v'_1 = v_2 \bmod (\ell_{r+1} + 1),$$

$$v'_2 = \left\lfloor \frac{v_2}{\ell_{r+1} + 1} \right\rfloor + v_1 \prod_{\substack{m=1, \\ m \neq s, r}}^n (\ell_m + 1).$$

The Bernstein patch $B(\mathbf{u})$ arranged accordingly in the $(\ell_1 + 1) \times \ell^*$ Bernstein matrix, denoted by $\mathcal{B}(\mathbf{u})$, is given by Λ_n .

Matrix Method for a General Box. Firstly, we affinely map a given box \mathbf{x} ,

$$\mathbf{x} = ([\underline{x}_s, \bar{x}_s])_{s=1}^n, \text{ with } \underline{x}_s < \bar{x}_s, s = 1, \dots, n, \tag{13}$$

to the unit box \mathbf{u} by

$$z_s = \frac{x_s - \underline{x}_s}{\bar{x}_s - \underline{x}_s}, \quad s = 1, \dots, n. \tag{14}$$

We sequentially transform x_s , $s = 1, \dots, n$. By substituting (14) in (2) for one x_s , $s = 1, \dots, n$, at a time, we obtain a polynomial p^* over \mathbf{u} . The coefficients of p^* arranged in an $(\ell_1 + 1) \times \ell^*$ matrix, say A^* , can be derived as follows from the matrix A of the coefficients of p given in (8): For $s = 1, \dots, n$ define

$$Q_s := \begin{cases} \tilde{D}_s(\frac{\bar{x}_s - x_s}{x_s}) P_s^T \tilde{D}_s(x_s), & x_s \neq 0, \\ \tilde{D}_s(\bar{x}_s), & x_s = 0, \end{cases} \tag{15}$$

where $\tilde{D}_s(t)$ is the diagonal matrix of order $\ell_s + 1$

$$\tilde{D}_s(t) := \text{diag}(1, t, t^2, \dots, t^{\ell_s}), \quad s = 1, \dots, n.$$

Then A^* can be represented as

$$A^* = (Q_n(\dots(Q_2(Q_1 A)^c \dots)^c)^c. \tag{16}$$

By applying the procedure for the unit box to the matrix A^* , we obtain the Bernstein patch of p over \mathbf{x} .

Amount of Arithmetic Operations. Assuming that $\kappa = \ell_s$ for all $s = 1, \dots, n$, the presented matrix method requires $n\kappa \frac{(\kappa+1)^n}{2}$ additions and $n(\kappa+1)^n$ multiplications for the computation of the Bernstein coefficients over the unit box \mathbf{u} , and needs $n\kappa(\kappa+1)^n + n$ additions and $3n(\kappa+1)^n + 2n(\kappa-1) + n$ multiplications for a general box. A verified version of this method which is taking into account of all rounding errors as well as data uncertainties was implemented by Dr. Florian Bünger, Hamburg University of Technology, Germany. It is included in the version 12 of the MATLAB toolbox INTLAB [11].

4 Computation of the Bernstein Coefficients of the Product of Two Multivariate Polynomials

Let p and q be two n -variate polynomials of degree $\ell(p)$ and $\ell(q)$, respectively, with the Bernstein expansions of degrees $d(p) \geq \ell(p)$ and $d(q) \geq \ell(q)$ over \mathbf{x}

$$p(x) = \sum_{j=0}^{d(p)} b_j(p) B_j^{(d(p))}(x), \tag{17a}$$

$$q(x) = \sum_{i=0}^{d(q)} b_i(q) B_i^{(d(q))}(x). \tag{17b}$$

For the ease of presentation we consider here only the unit box \mathbf{u} and assume that $d(p) = \ell(p)$ and $d(q) = \ell(q)$. Then, the polynomial pq resulting when multiplying p and q is of degree $\ell = \ell(p) + \ell(q)$. Hence, the Bernstein representation of pq over \mathbf{u} is given as

$$pq(x) = \sum_{m=0}^{\ell} b_m(pq) B_m^{(\ell)}(x), \tag{18}$$

where $b_m(pq)$ is the m -th Bernstein coefficient of pq of degree ℓ over \mathbf{u} , $m = 0, \dots, \ell$. Let $B(p, \mathbf{u})$, $B(q, \mathbf{u})$, and $B(pq, \mathbf{u})$ denote the Bernstein patches of p , q , and pq over \mathbf{u} , respectively, and their corresponding Bernstein matrices are given by $\mathcal{B}(p, \mathbf{u})$, $\mathcal{B}(q, \mathbf{u})$ and $\mathcal{B}(pq, \mathbf{u})$. In this section, we present two matrix methods, which are named the *first method* and *second method*, for the computation of the Bernstein coefficients $b_m(pq)$ of pq .

4.1 First Method

By this method the Bernstein coefficients of pq are computed from the Bernstein representation of p and q . The representations (17a) and (17b) can be rewritten as

$$p(x) = (1-x)^{\ell(p)} \sum_{j=0}^{\ell(p)} c_j(p) \left(\frac{x}{1-x} \right)^j, \tag{19a}$$

$$q(x) = (1-x)^{\ell(q)} \sum_{i=0}^{\ell(q)} c_i(q) \left(\frac{x}{1-x} \right)^i, \tag{19b}$$

where $c_j(p)$ and $c_i(q)$ are called the *scaled Bernstein coefficients* of p and q which are given by

$$c_j(p) = b_j(p) \binom{\ell(p)}{j}, \quad j = 0, \dots, \ell(p), \tag{20a}$$

$$c_i(q) = b_i(q) \binom{\ell(q)}{i}, \quad i = 0, \dots, \ell(q). \tag{20b}$$

From (19a) and (19b), the power representation of pq is obtained as

$$pq(x) = (1 - x)^\ell \sum_{m=0}^{\ell} c_m \left(\frac{x}{1 - x} \right)^m,$$

where the m -th scaled Bernstein coefficient c_m of pq is given as

$$c_m = \sum_{\mu=0}^m c_\mu(p)c_{m-\mu}(q), \quad m = 0, \dots, \ell, \tag{21}$$

with $c_\mu(p) := 0$ if $\mu_s > \ell_s(p)$ and $c_\mu(q) := 0$ if $\mu_s > \ell_s(q)$ for some $s \in \{1, \dots, n\}$. Then, from (21) the m -th Bernstein coefficient of pq is

$$b_m(pq) = \frac{c_m}{\binom{\ell}{m}}, \quad 0 \leq m \leq \ell. \tag{22}$$

In matrix language, the first method can be described as follows: let for $s = 1, \dots, n$, the diagonal matrix D_s of order $\ell_s(p) + 1$ be defined by

$$D_s := \text{diag} \left(\binom{\ell_s(p)}{0}, \binom{\ell_s(p)}{1}, \dots, \binom{\ell_s(p)}{\ell_s(p)} \right), \tag{23}$$

and let $C(p)$ be the $(\ell_1(p) + 1) \times \ell^*(p)$ matrix, where $\ell^*(p) := \prod_{s=2}^n (\ell_s(p) + 1)$, which is obtained by

$$C(p) = (D_n(\dots(D_2(D_1\mathcal{B}(p, \mathbf{u}))^c)\dots)^c)^c. \tag{24}$$

It is easy to see that $(C(p))_{q_1, q_2} = c_j(p)$, $q_1 = 0, \dots, \ell_1(p)$, $q_2 = 0, \dots, \ell^*(p) - 1$ and $j = 0, \dots, \ell(p)$, where the correspondence between the scaled Bernstein coefficient $c_j(p)$ and the entries of $C(p)$ can be determined by using (7).

Let us define for $s = 1, \dots, n$, $t = 0, \dots, \ell_s(q)$, the following $(\ell_s + 1) \times (\ell_s(p) + 1)$ matrices $W_s^{(t)}$ row-wise by

$$\begin{aligned} W_s^{(t)} [0, \dots, t - 1] &:= 0, \\ W_s^{(t)} [t, \dots, \ell_s(p) + t] &:= I_{\ell_s(p)+1} \text{ (identity matrix of order } \ell_s(p) + 1), \\ W_s^{(t)} [\ell_s(p) + t + 1, \dots, \ell_s] &:= 0; \end{aligned}$$

as a convention, we define for $t = 0, \ell_s(q)$

$$\begin{aligned} W_s^{(t)} [0, \dots, -1] &= \phi, \\ W_s^{(t)} [\ell_s + 1, \dots, \ell_s] &= \phi, \end{aligned}$$

where ϕ is a matrix of size 0×0 . Assume that the scaled Bernstein coefficients of pq , see (21), are arranged in the $(\ell_1 + 1) \times \ell^*$ matrix $C(pq)$, with $\ell^* = \prod_{s=2}^n (\ell_s + 1)$ that is given by

$$C(pq) = \sum_{i_n=0}^{\ell_n(q)} \dots \sum_{i_1=0}^{\ell_1(q)} \left(W_n^{(i_n)} \left(\dots \left(W_2^{(i_2)} \left(W_1^{(i_1)} c_{i_1, \dots, i_n}(q) C(p) \right)^c \dots \right)^c \right)^c \tag{25}$$

such that $(C(pq))_{q_1, q_2} = c_m$, $q_1 = 0, \dots, \ell_1$, $q_2 = 0, \dots, \ell^* - 1$, and $m = 0, \dots, \ell$. Here, the relation between the entries of $C(pq)$ and the scaled Bernstein coefficients of pq , c_m , $m = 0, \dots, \ell$, can be determined as in (7). From (25), we get

$$\mathcal{B}(pq, \mathbf{u}) = \left(D'_n \left(\dots \left(D'_2 \left(D'_1 C(pq) \right)^c \dots \right)^c \right)^c, \tag{26}$$

where D'_s is the inverse of the diagonal matrix $\text{diag} \left(\binom{\ell_s}{0}, \binom{\ell_s}{1}, \dots, \binom{\ell_s}{\ell_s} \right)$ for $s = 1, \dots, n$. A pseudocode for the first method is given in Algorithm 1. Its performance is illustrated in Appendix A.

4.2 Second Method

Let $a_j(p)$ and $a_i(q)$ be the j -th and the i -th coefficients of the power representations of p and q , respectively, such that $j = 0, \dots, \ell(p)$ and $i = 0, \dots, \ell(q)$. Assume that $a_j(p)$ are arranged in an $(\ell_1(p) + 1) \times \ell^*(p)$ matrix $A(p)$. Recall that pq is an n -variate polynomial of degree ℓ . Then the power representation of pq is given by

$$pq(x) = \sum_{m=0}^{\ell} a_m x^m. \tag{27}$$

We arrange the coefficients of pq in an $(\ell_1 + 1) \times \ell^*$ matrix \hat{A} . In this method, the computation of the Bernstein coefficients of pq , see (18), is based on its power representation (27). The matrix description of this method is as follows: the entries of \hat{A} are the entries of the matrix that is obtained from (25), where the (r_1, r_2) -th entry of $C(p)$ is replaced by the (r_1, r_2) -th entry of $A(p)$, where $r_1 = 0, \dots, \ell_1(p)$, $r_2 = 0, \dots, \ell^*(p) - 1$, and $c_i(q)$ is replaced by $a_i(q)$, $0 \leq i \leq \ell(q)$. Then the method presented in Subsect. 3.2 is applied to compute the Bernstein coefficients of pq starting from \hat{A} .

4.3 Amount of Arithmetic Operations

In Tables 1 and 2, the number of the arithmetic operations of both methods are presented. For simplicity, we assume that $\ell_s(p) = \ell_s(q) = \kappa$ for $s = 1, \dots, n$. Furthermore, we use the method from Subsect. 3.2 for the computation of $B(p, \mathbf{u})$ and $B(q, \mathbf{u})$ in the first method, see (24), and $B(pq, \mathbf{u})$ in the second method.

For the ease of comparison, we assume that the basis operations (addition, multiplication, and division) are taking the same time. Then we conclude that for $n \geq 4$ and $\kappa = 1$, $n \geq 2$ and $\kappa = 2, 3$, and for all n and $\kappa \geq 4$ the first method is superior to the second method. In addition, the first method has the advantage that all computations are performed *ab initio* in the Bernstein representation so that the numerical stability of this representation with respect to perturbations of initial data, or rounding errors that occur during floating point calculations can be fully employed, see [6, Section 6].

Algorithm 1. First method for the computation of the Bernstein coefficients of pq over the unit box \mathbf{u}

- 1: *Input: The coefficients of the power representation of p and q over \mathbf{u}*
 - 2: *Output: The matrix $\mathcal{B}(pq, \mathbf{u})$ containing the Bernstein coefficients of pq over \mathbf{u}*
 - 3: *Step 1: Compute the Bernstein coefficients of p, q and arrange them in matrices*
 - 4: $\mathcal{B}(p, \mathbf{u})$ and $\mathcal{B}(q, \mathbf{u})$, respectively, by using the method from Subsection 3.2.
 - 5: *Step 2: Compute $C(p)$.*
 - 6: Put $C_0(p) := \mathcal{B}(p, \mathbf{u})$.
 - 7: **for** $s = 1, \dots, n$ **do**
 - 8: Compute $C_s(p) := (D_s C_{s-1}(p))^c$.
 - 9: **end for**
 - 10: Put $C_0(pq) := O$.
 - 11: *Step 3: Computation of $C(pq)$*
 - 12: **for** $i_1 = 0, \dots, \ell_1(q)$ **do**
 - 13: \vdots
 - 14: **for** $i_n = 0, \dots, \ell_n(q)$ **do**
 - 15: Put $M_0^{(i)} := c_i(q)C_n(p)$, where $c_i(q)$ is given by (20b).
 - 16: **for** $r = 1, \dots, n$ **do**
 - 17: $M_r^{(i)} := (W_r^{(i_r)} M_{r-1}^{(i)})^c$.
 - 18: **end for**
 - 19: $C_0(pq) := M_n^{(i)} + C_0(pq)$.
 - 20: **end for**
 - 21: \vdots
 - 22: **end for**
 - 23: *Step 4: Computation of $\mathcal{B}(pq, \mathbf{u})$*
 - 24: Put $F_0 := C_0(pq)$.
 - 25: **for** $s = 1, \dots, n$ **do**
 - 26: Compute $F_s := (D'_s F_{s-1})^c$, see (26).
 - 27: **end for**
 - 28: Put $\mathcal{B}(pq, \mathbf{u}) := F_n$.
 - 29: *Step 5: End of the algorithm*
-

Table 1. Number of real arithmetic operations required to obtain $B(pq, \mathbf{u})$ by the first method

Calculation of	Number of additions	Number of multi- plications/divisions
$B(p, \mathbf{u})$ and $B(q, \mathbf{u})$ by the method in Subsect. 3.2	$n\kappa(\kappa + 1)^n$	$2n(\kappa + 1)^n$
$C(p)$	0	$n(\kappa + 1)^n$
$C(pq)$	$(\kappa + 1)^n [(\kappa + 1)^n - 1] - n\kappa(\kappa + 1)^{n-1}$	$(\kappa + 1)^{2n}$
$B(pq, \mathbf{u})$	0	$n(2\kappa + 1)^n$

Table 2. Number of arithmetic operations to obtain $B(pq, \mathbf{u})$ by the second method

Calculation of	Number of additions	Number of multi- plications/divisions
$C(pq)$	$(\kappa + 1)^n[(\kappa + 1)^n - 1] - n\kappa(\kappa + 1)^{n-1}$	$(\kappa + 1)^{2n}$
$B(pq, \mathbf{u})$ by the method in Subsect. 3.2	$n\kappa(2\kappa + 1)^n$	$n(2\kappa + 1)^n$

5 Matrix Method for the Computation of the Bernstein Coefficients of a Multivariate Polynomial from Those of One of Its Partial Derivatives

Let p be an n -variate polynomial of degree ℓ with the power representation given as in (2). Assume that its coefficients are arranged in the matrix A , which is presented in (8). We expand p into Bernstein polynomials of degree d , $d \geq \ell$, over the box \mathbf{x} (13) as in (3). Without loss of generality, we assume that $d = \ell$. Recall that the Bernstein representation of the first partial derivative of p with respect to x_s , $s \in \{1, \dots, n\}$, is given by

$$\frac{\partial p}{\partial x_s} = \sum_{i \leq \ell_{s,-1}} \tilde{b}_i^{(\ell_{s,-1})} B_i^{(\ell_{s,-1})}(\mathbf{x}), \tag{28}$$

where for $j = 0, \dots, \ell_{s,-1}$

$$\tilde{b}_j^{(\ell_{s,-1})} = \ell_s (b_{j_{s,1}}^{(\ell)} - b_j^{(\ell)}) \tag{29}$$

denotes the j -th Bernstein coefficient of $\frac{\partial p}{\partial x_s}$ of degree $\ell_{s,-1}$ over \mathbf{x} , i.e., the Bernstein coefficients of $\frac{\partial p}{\partial x_s}$ can be obtained from differences between its successive Bernstein coefficients, e.g., [9, formula (4)].

Assume that the Bernstein coefficients of $\frac{\partial p}{\partial x_s}$, $s \in \{1, \dots, n\}$, are given and are arranged in the Bernstein patch $B(\frac{\partial p}{\partial x_s}, \mathbf{x})$. In the following, we present a matrix method by which the Bernstein patch $B(p, \mathbf{x})$ that comprises the Bernstein coefficients of p over \mathbf{x} can be computed using $B(\frac{\partial p}{\partial x_s}, \mathbf{x})$. Without loss of generality, we assume that $s = 1$. From (29), it follows that

$$b_{i_{1,1}}^{(\ell)} = \frac{\tilde{b}_i^{(\ell_{1,-1})}}{\ell_1} + b_i^{(\ell)}, \quad i = 0, \dots, \ell_{1,-1}. \tag{30}$$

In other words, for computing $B(p, \mathbf{x})$ it is sufficient to compute the Bernstein coefficients $b_{i_{1,0}}^{(\ell)}$, see (6), then the remaining coefficients can be obtained iteratively using (30). The coefficients $b_{i_{1,0}}^{(\ell)}$ are the Bernstein coefficients of p for $x_1 = \underline{x}_1$. By the face value property of the Bernstein coefficients, see Subsect. 3.1, these coefficients are identical to those that are located at the corresponding $(n - 1)$ -dimensional face of $B(p, \mathbf{x})$; they are obtained from $B(p, \mathbf{x})$ by firstly

freezing p on the face of \mathbf{x} with $x_1 = \underline{x}_1$ and then computing the Bernstein coefficients of the resulting polynomial. In matrix language, the computation is as follows. Denote by C_1 the row vector of length ℓ^* , where $\ell^* = \prod_{s=2}^n (\ell_s + 1)$, that contains the coefficients of p such that $x_1 = \underline{x}_1$. For $\mu = 1, \dots, \ell_1$, we define the elementary bidiagonal matrices $H_\mu(x) \in \mathbb{R}^{\mu, \mu+1}$ by

$$(H_\mu(x))_{i,j} := \begin{cases} 1, & i = j, \\ x, & i = \mu, j = \mu + 1, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

Then, C_1 can be obtained as

$$C_1 = \overline{H}_1(\underline{x}_1) \cdots H_{\ell_1-1}(\underline{x}_1) H_{\ell_1}(\underline{x}_1) A. \tag{32}$$

From C_1 , we define the $(\ell_2 + 1) \times \prod_{r=3}^n (\ell_r + 1)$ matrix A_1 with coefficients given for $q_1 = 0, \dots, \ell_2$ and $q_2 = 0, \dots, \prod_{r=3}^n (\ell_r + 1) - 1$ by

$$(A_1)_{q_1, q_2} := (C_1)_{q_1 + q_2(\ell_2 + 1) + 1}. \tag{33}$$

Then the method from Subsect. 3.2 for the computation of the Bernstein coefficients of p on the face of \mathbf{x} with $x_1 = \underline{x}_1$ starting from A_1 is applied. We denote the resulting matrix by B_1 and arrange its entries in the row vector C'_1 of length ℓ^* , such that for $r_1 = 1, \dots, \ell^*$, $v_1 = 0, \dots, \ell_2$, and $v_2 = 0, \dots, \prod_{r=3}^n (\ell_r + 1) - 1$, we have

$$(C'_1)_{r_1} = (B_1)_{v_1, v_2}, \tag{34}$$

where

$$r_1 = v_1 + v_2(\ell_2 + 1) + 1.$$

Let B'_1 be the $(\ell_1 + 1) \times \ell^*$ matrix defined for $r_1 = 0, \dots, \ell_1$ and $r_2 = 0, \dots, \ell^* - 1$ by

$$(B'_1)_{r_1, r_2} = \begin{cases} (C'_1)_{r_2 + 1}, & \text{if } r_1 = 0, \\ \left(\frac{1}{\ell_1} \mathcal{B}\left(\frac{\partial p}{\partial x_s}, \mathbf{x}\right)\right)_{r_1 - 1, r_2}, & \text{if } r_1 = 1, \dots, \ell_1. \end{cases} \tag{35}$$

For $\mu = 1, \dots, \ell_1$, we define the square matrices $H_\mu^{(1)}$ of order $\ell_1 + 1$, such that for $r_v = 0, \dots, \ell_1, v = 1, 2$, it is given by

$$(H_\mu^{(1)})_{r_1, r_2} := \begin{cases} 1, & r_1 = r_2, \\ 1, & r_1 = \ell_1 - \mu + 1, r_2 = \ell_1 - \mu, \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

Then, the Bernstein matrix $\mathcal{B}(p, \mathbf{x})$ that comprises the Bernstein coefficients of p of degree ℓ over \mathbf{x} , can be calculated by

$$\mathcal{B}(p, \mathbf{x}) = H_1^{(1)} \cdots H_{\ell_1-1}^{(1)} H_{\ell_1}^{(1)} B'_1, \tag{37}$$

where the correspondence between $\mathcal{B}(p, \mathbf{x})$ and $B(p, \mathbf{x})$ can be determined by using (7). As a consequence of our initial assumption, for computing $B(p, \mathbf{x})$ from $B(\frac{\partial p}{\partial x_s}, \mathbf{x})$, where $s \in \{2, \dots, n\}$, we firstly employ the cyclic ordering with respect to x_s , in such a way that the multiplications in (32) and (37) are well defined.

In Table 3, the number of arithmetic operations needed for the computation of $\mathcal{B}(p, \mathbf{x})$ is presented. For simplicity, we assume here that $\ell_s = \kappa$, $s = 1, \dots, n$.

Table 3. Number of real arithmetic operations required to obtain $\mathcal{B}(p, \mathbf{x})$ from one of the partial derivatives of p

Calculation of	Number of additions	Number of multiplications/divisions
C_1 by (32)	$\kappa(\kappa + 1)^{n-1}$	$\kappa(\kappa + 1)^{n-1}$
B_1 over a general box \mathbf{x}' which is obtained from \mathbf{x} by freezing $x_1 = x_1$ using the method in Subsect. 3.2	$(n - 1)\kappa(\kappa + 1)^{n-1} + n - 1$	$3(n - 1)(\kappa + 1)^{n-1} + 2(n - 1)(\kappa - 1) + n - 1$
B'_1 by (35)	0	$\kappa(\kappa + 1)^{n-1}$
$\mathcal{B}(p, \mathbf{x})$ by (37)	$\kappa(\kappa + 1)^{n-1}$	0

In total, the computation of $\mathcal{B}(p, \mathbf{x})$ requires $(n + 1)\kappa(\kappa + 1)^{n-1} + n - 1$ additions and $2\kappa(\kappa + 1)^{n-1} + 3(n - 1)(\kappa + 1)^{n-1} + 2(n - 1)(\kappa - 1) + n - 1$ multiplications.

Appendix A. Example for the Performance of Algorithm 1

Let p and q be bivariate polynomials of degree (4, 2) and (3, 2), respectively, with Bernstein matrices over \mathbf{u}

$$\mathcal{B}(p, \mathbf{u}) = \begin{bmatrix} 2 & 0 & -1 \\ -2 & 3 & 0 \\ 1 & 3 & -3 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{and} \quad \mathcal{B}(q, \mathbf{u}) = \begin{bmatrix} -1 & -4 & -1 \\ 2 & 0 & 5 \\ -2 & 3 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Then

$$D_1 = \text{diag}(1, 4, 6, 4, 1) \text{ and } D_2 = \text{diag}(1, 2, 1),$$

$$C_1(p) = (D_1 C_0(p))^c = \begin{bmatrix} 2 & -8 & 6 & 4 & 0 \\ 0 & 12 & 18 & 0 & -1 \\ -1 & 0 & -18 & 4 & 0 \end{bmatrix},$$

$$C_2(p) = (D_2 C_1(p))^c = \begin{bmatrix} 2 & 0 & -1 \\ -8 & 24 & 0 \\ 6 & 36 & -18 \\ 4 & 0 & 4 \\ 0 & -2 & 0 \end{bmatrix};$$

$$M_0^{(1,1)} = M_0^{(2,2)} = M_0^{(3,0)} = O_{5,3},$$

$$M_0^{(0,0)} = \begin{bmatrix} -2 & 0 & 1 \\ 8 & -24 & 0 \\ -6 & -36 & 18 \\ -4 & 0 & -4 \\ 0 & 2 & 0 \end{bmatrix}, M_0^{(0,1)} = \begin{bmatrix} -16 & 0 & 8 \\ 64 & -192 & 0 \\ -48 & -288 & 144 \\ -32 & 0 & -32 \\ 0 & 16 & 0 \end{bmatrix}, M_0^{(0,2)} = \begin{bmatrix} -2 & 0 & 1 \\ 8 & -24 & 0 \\ -6 & -36 & 18 \\ -4 & 0 & -4 \\ 0 & 2 & 0 \end{bmatrix},$$

$$M_0^{(1,0)} = \begin{bmatrix} 12 & 0 & -6 \\ -48 & 144 & 0 \\ 36 & 216 & -108 \\ 24 & 0 & 24 \\ 0 & -12 & 0 \end{bmatrix}, M_0^{(1,2)} = \begin{bmatrix} 30 & 0 & -15 \\ -120 & 360 & 0 \\ 90 & 540 & -270 \\ 60 & 0 & 60 \\ 0 & -30 & 0 \end{bmatrix},$$

$$M_0^{(2,0)} = \begin{bmatrix} -12 & 0 & 6 \\ 48 & -144 & 0 \\ -36 & -216 & 108 \\ -24 & 0 & -24 \\ 0 & 12 & 0 \end{bmatrix}, M_0^{(2,1)} = \begin{bmatrix} 36 & 0 & -18 \\ -144 & 432 & 0 \\ 108 & 648 & -324 \\ 72 & 0 & 72 \\ 0 & -36 & 0 \end{bmatrix},$$

$$M_0^{(3,1)} = \begin{bmatrix} 4 & 0 & -2 \\ -16 & 48 & 0 \\ 12 & 72 & -36 \\ 8 & 0 & 8 \\ 0 & -4 & 0 \end{bmatrix}, M_0^{(3,2)} = \begin{bmatrix} 2 & 0 & -1 \\ -8 & 24 & 0 \\ 6 & 36 & -18 \\ 4 & 0 & 4 \\ 0 & -2 & 0 \end{bmatrix};$$

$$M_1^{(0,0)} = \begin{bmatrix} -2 & 8 & -6 & -4 & 0 & 0 & 0 & 0 \\ 0 & -24 & -36 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 18 & -4 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad M_2^{(0,0)} = \begin{bmatrix} -2 & 0 & 1 & 0 & 0 \\ 8 & -24 & 0 & 0 & 0 \\ -6 & -36 & 18 & 0 & 0 \\ -4 & 0 & -4 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M_1^{(0,1)} = \begin{bmatrix} -16 & 64 & -48 & -32 & 0 & 0 & 0 & 0 \\ 0 & -192 & -288 & 0 & 16 & 0 & 0 & 0 \\ 8 & 0 & 144 & -32 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad M_2^{(0,1)} = \begin{bmatrix} 0 & -16 & 0 & 8 & 0 \\ 0 & 64 & -192 & 0 & 0 \\ 0 & -48 & -288 & 144 & 0 \\ 0 & -32 & 0 & -32 & 0 \\ 0 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$M_1^{(0,2)} = \begin{bmatrix} -2 & 8 & -6 & -4 & 0 & 0 & 0 & 0 \\ 0 & -24 & -36 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 18 & -4 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad M_2^{(0,2)} = \begin{bmatrix} 0 & 0 & -2 & 0 & 1 \\ 0 & 0 & 8 & -24 & 0 \\ 0 & 0 & -6 & -36 & 18 \\ 0 & 0 & -4 & 0 & -4 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The remaining 18 matrices $M_s^{(i)}$, $s = 1, 2$, are formed analogously.

$$F_0 = C_0(pq) = \begin{bmatrix} -2 & -16 & -1 & 8 & 1 \\ 20 & 40 & -160 & -24 & -15 \\ -66 & 96 & -390 & 450 & 18 \\ 80 & -100 & 408 & 506 & -275 \\ -12 & -122 & 896 & -298 & 60 \\ -24 & 72 & 54 & 42 & -18 \\ 0 & 20 & -32 & 8 & 4 \\ 0 & 0 & 4 & -2 & 0 \end{bmatrix};$$

$$D'_1 = \text{diag}(1, \frac{1}{7}, \frac{1}{21}, \frac{1}{35}, \frac{1}{35}, \frac{1}{21}, \frac{1}{7}, 1), \quad D'_2 = \text{diag}(1, \frac{1}{4}, \frac{1}{6}, \frac{1}{4}, 1).$$

The Bernstein matrix of the product of the polynomials p and q is given by

$$\mathcal{B}(pq, \mathbf{u}) = F_2 = (D'_2(D'_1 F_0)^c)^c = \begin{bmatrix} -2 & -4 & \frac{-1}{6} & 2 & 1 \\ \frac{20}{7} & \frac{10}{7} & \frac{-80}{21} & \frac{-6}{7} & \frac{-15}{7} \\ \frac{-22}{7} & \frac{8}{7} & \frac{-65}{21} & \frac{75}{14} & \frac{6}{7} \\ \frac{16}{7} & \frac{-5}{7} & \frac{68}{35} & \frac{253}{70} & \frac{-55}{7} \\ \frac{-12}{35} & \frac{-61}{70} & \frac{64}{15} & \frac{-149}{70} & \frac{12}{7} \\ \frac{-8}{7} & \frac{6}{7} & \frac{3}{7} & \frac{1}{2} & \frac{-6}{7} \\ 0 & \frac{5}{7} & \frac{-16}{21} & \frac{2}{7} & \frac{4}{7} \\ 0 & 0 & \frac{2}{3} & \frac{-1}{2} & 0 \end{bmatrix}.$$

Appendix B. Example for the Performance of the Method Presented in Sect. 5

Let $p(x_1, x_2) = -504x_1^4x_2^2 - 84x_1^4x_2 + 288x_1^4 + 6x_1^3x_2^2 + 30x_1^3x_2 - 60x_1^3 + 36x_1^2x_2^2 - 20x_1^2x_2 + 28x_1^2 - 54x_1x_2^2 + 21x_1x_2 - 24x_1 + 24x_2^2 - 24x_2 + 48$. Then

$$\mathcal{B}\left(\frac{\partial p}{\partial x_1}, \mathbf{u}\right) = \begin{bmatrix} -24 & \frac{-27}{2} & -57 \\ -4 & \frac{-9}{6} & \frac{-83}{3} \\ \frac{-140}{3} & \frac{-207}{6} & \frac{-67}{6} \\ 1004 & \frac{-1743}{2} & -1241 \end{bmatrix} \text{ and } C'_1 = \mathcal{B}(p(0, x_2), [0, 1]) = [48 \ 36 \ 48];$$

$$B'_1 = \begin{bmatrix} 48 & 36 & 48 \\ -6 & \frac{-27}{8} & \frac{-57}{4} \\ \frac{-4}{3} & \frac{-9}{24} & \frac{-83}{12} \\ \frac{-140}{12} & \frac{-207}{24} & \frac{-67}{12} \\ 251 & \frac{-1743}{8} & \frac{-1241}{4} \end{bmatrix}.$$

The Bernstein coefficients of p over \mathbf{u} are obtained from

$$\begin{aligned} \mathcal{B}(p, \mathbf{u}) &= H_1 H_2 H_3 H_4 B'_1 \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 48 & 36 & 48 \\ -6 & \frac{-27}{8} & \frac{-57}{4} \\ \frac{-4}{3} & \frac{-9}{24} & \frac{-83}{12} \\ \frac{-140}{12} & \frac{-207}{24} & \frac{-67}{12} \\ 251 & \frac{-1743}{8} & \frac{-1241}{4} \end{bmatrix} \\ &= \begin{bmatrix} 48 & 36 & 48 \\ 42 & \frac{261}{8} & \frac{135}{4} \\ \frac{122}{3} & \frac{774}{24} & \frac{322}{12} \\ 29 & \frac{567}{24} & \frac{255}{12} \\ 280 & \frac{483}{2} & -289 \end{bmatrix}. \end{aligned}$$

References

1. Alonso, P., Delgado, J., Gallego, R., Peña, J.M.: Conditioning and accurate computations with Pascal matrices. *J. Comput. Appl. Math.* **252**, 21–26 (2013)
2. Clauss, P., Chupaeva, I.Y.: Application of symbolic approach to the Bernstein expansion for program analysis and optimization. In: Duesterwald, E. (ed.) *Compiler Construction*. LNCS, vol. 2985, pp. 120–133. Springer, Berlin, Heidelberg (2004)
3. Clauss, P., Fernández, F.J., Garbervetsky, D., Verdoolaege, S.: Symbolic polynomial maximization over convex sets and its application to memory requirement estimation. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **17**(8), 983–996 (2009)
4. Dang, T., Dreossi, T., Fanchon, É., Maler, O., Piazza, C., Rocca, A.: Set-based analysis for biological modelling. In: Liò, P., Zuliani, P. (eds.) *Automated Reasoning for Systems Biology and Medicine*. COBO, vol. 30, pp. 157–189. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-17297-8>
5. Dreossi, T.: Sapo: Reachability computation and parameter synthesis of polynomial dynamical systems. In: *Proceedings of International Conference Hybrid Systems: Computation and Control*, pp. 29–34, ACM, New York (2017)
6. Farouki, R.T.: The Bernstein polynomial basis: A centennial retrospective. *Comput. Aided Geom. Design* **29**, 379–419 (2012)
7. Farouki, R.T., Rajan, V.T.: Algorithms for polynomials in Bernstein form. *Comput. Aided Geom. Design* **5**, 1–26 (1988)
8. Garloff, J.: Convergent bounds for the range of multivariate polynomials. In: Nickel, K. (ed.) *Interval Mathematics 1985*. LNCS, vol. 212, pp. 37–56. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-16437-5_5
9. Garloff, J., Smith, A.P.: Solution of systems of polynomial equations by using Bernstein expansion. In: Alefeld, G., Rump, S., Rohn, J., and Yamamoto J. (eds.), *Symbolic Algebraic Methods and Verification Methods*, pp. 87–97. Springer (2001)
10. Rivlin, T.J.: Bounds on a polynomial. *J. Res. Nat. Bur. Standards* **74**(B), 47–54 (1970)
11. Rump, S.M.: INTLAB-INTerval LABoratory. In: Csendes, T. (ed.) *Developments in Reliable Computing*, pp. 77–104. Kluwer Academic Publishers, Dordrecht (1999)
12. Smith, A.P.: Fast construction of constant bound functions for sparse polynomials. *J. Global Optim.* **43**, 445–458 (2009)
13. Titi, J.: Matrix methods for the tensorial and simplicial Bernstein forms with application to global optimization, dissertation. University of Konstanz, Konstanz, Germany (2019). Available at <https://nbn-resolving.de/urn:nbn:de:bsz:352-2-k106crqmste71>
14. Titi, J., Garloff, J.: Fast determination of the tensorial and simplicial Bernstein forms of multivariate polynomials and rational functions. *Reliab. Comput.* **25**, 24–37 (2017)
15. Titi, J., Garloff, J.: Matrix methods for the simplicial Bernstein representation and for the evaluation of multivariate polynomials. *Appl. Math. Comput.* **315**, 246–258 (2017)
16. Titi, J., Garloff, J.: Matrix methods for the tensorial Bernstein form. *Appl. Math. Comput.* **346**, 254–271 (2019)
17. Titi, J., Garloff, J.: Bounds for the range of a complex polynomial over a rectangular region, submitted (2020)