Theoretical Explanation of Bernstein Polynomials' Efficiency: They Are Optimal Combination of Optimal Endpoint-Related Functions*

J. Nava and V. Kreinovich
Department of Computer Science, University of Texas
at El Paso, El Paso, Texas 79968, USA
jenava@miners.utep.edu,vladik@utep.edu

Abstract

In many applications of interval computations, it turned out to be beneficial to represent polynomials on a given interval $[\underline{x}, \overline{x}]$ as linear combinations of *Bernstein* polynomials $(x - \underline{x})^k \cdot (\overline{x} - x)^{n-k}$. In this paper, we provide a theoretical explanation for this empirical success: namely, we show that under reasonable optimality criteria, Bernstein polynomials can be uniquely determined from the requirement that they are optimal combinations of optimal polynomials corresponding to the interval's endpoints.

Keywords: Bernstein polynomials, interval computations, symmetries, optimization
AMS subject classifications: 65G20 65G40 65K99 58J70

1 Formulation of the Problem

Polynomials are often helpful. In many areas of numerical analysis, in particular, in computations with automatic results verification, it turns out to be helpful to approximate a dependence by a polynomial. For example, in computations with automatic results verification, Taylor methods – in which the dependence is approximated by a polynomial – turned out to be very successful; see, e.g., [1, 2, 3, 8, 9, 11].

The efficiency of polynomials can be theoretically explained. The efficiency of polynomials is not only an empirical fact, this efficiency can also be theoretically justified. Namely, in [12], it was shown that under reasonable assumptions on the optimality criterion – like invariance with respect to selection a starting point and a measuring unit for describing a quantity – every function from the optimal class of approximating functions is a polynomial.

*Submitted: July 15, 2011; Revised: ?; Accepted: ?

How to represent polynomials in a computer: straightforward way and Bernstein polynomials. In view of the fact that polynomials are efficient, we need to represent them inside the computer. A straightforward way to represent a polynomial is to store the coefficients at its monomials. For example, a natural way to represent a quadratic polynomial $f(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2$ is to store the coefficients c_0, c_1, and c_2.

It turns out that in many applications in which we are interested in functions defined on a given interval $[\underline{x}, \overline{x}]$, we get better results if instead, we represent a general polynomial as a linear combination of *Bernstein polynomials*, i.e., functions proportional to $(x - \underline{x})^k \cdot (\overline{x} - x)^{n-k}$, and store coefficients of this linear combination; see, e.g., [4, 5, 6, 7, 10, 13].

For example, a general quadratic polynomial on the interval $[0, 1]$ can be represented as

$$f(x) = a_0 \cdot x^0 \cdot (1-x)^2 + a_1 \cdot x^1 \cdot (1-x)^1 + a_0 \cdot x^2 \cdot (1-x)^0 = a_0 \cdot x^2 + a_1 \cdot x \cdot (1-x) + a_2 \cdot (1-x)^2;$$

to represent a generic polynomial in a computer, we store the values a_0, a_1, and a_2. (To be more precise, we store values proportional to a_i.)

For polynomials of several variables defined on a box $[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$, we can use similarly multi-dimensional Bernstein polynomials which are proportional to $\prod\limits_{i=1}^{n} (x_i - \underline{x}_i)^{k_i} \cdot (\overline{x}_i - x_i)^{n-k_i}$.

Natural questions. Natural questions are:

- why is the use of these basic functions more efficient than the use of standard monomials $\prod\limits_{i=1}^{n} x_i^{k_i}$?
- are Bernstein polynomials the best or these are even better expressions?

Towards possible answers to these questions. To answer these questions, we take into account that in the 1-D case, an interval $[\underline{x}, \overline{x}]$ is uniquely determined by its endpoints $\underline{x}$ and $\overline{x}$. Similarly, in the multi-D case, a general box $[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$ is uniquely determined by two multi-D "endpoints" $\underline{x} = (\underline{x}_1, \ldots \underline{x}_n)$ and $\overline{x} = (\overline{x}_1, \ldots, \overline{x}_n)$. It is therefore reasonable to design the basic polynomials as follows:

- first, we find two polynomial functions $\underline{f}(x)$ and $\overline{f}(x)$, where $x = (x_1, \ldots, x_n)$, related to each of the endpoints;
- then, we use some combination operation $F(a, b)$ to combine the functions $\underline{f}(x)$ and $\overline{f}(x)$ into a single function $f(x) = G(\underline{f}(x), \overline{f}(x))$.

In this paper, we use the approach from [12] to prove that if we select the optimal polynomials $\underline{f}(x)$ and $\overline{f}(x)$ on the first stage and the optimal combination operation on the second stage, then the resulting function $f(x)$ is proportional to a Bernstein polynomial.

In other words, we prove that under reasonable optimality criteria, Bernstein polynomials can be uniquely determined from the requirement that they are optimal combinations of optimal polynomials corresponding to the interval's endpoints.

2 Optimal Functions Corresponding to Endpoints: Towards a Precise Description of the Problem

Formulation of the problem: reminder. Let us first find optimal polynomials corresponding to endpoints $x^{(0)} = \underline{x}$ and $x^{(0)} = \overline{x}$.

We consider applications in which the dependence of a quantity y on the input values $x_1, \ldots, x_n$ is approximated by a polynomial $y = f(x) = f(x_1, \ldots, x_n)$. For each of the two endpoints $x^{(0)} = \underline{x}$ and $x^{(0)} = \overline{x}$, out of all polynomials which are "related" to this point, we want to find the one which is, in some reasonable sense, optimal.

How to describe this problem in precise terms. To describe this problem in precise terms, we need to describe:

- what it means for a polynomial to be "related" to the point, and
- what it means for one polynomial to be "better" than the other.

Physical meaning. To formalize the two above notions, we take into account that in many practical applications, the inputs numbers x_i are values of some physical quantities, and the output y also represent the value of some physical quantity.

Scaling and shift transformations. The numerical value of each quantity depends on the choice of a measuring unit and on the choice of the starting point. If we replace the original measuring unit by a unit which is λ times smaller (e.g., use centimeters instead of meters), then instead of the original numerical value y, we get a new value $y' = \lambda \cdot y$.

Similarly, if we replace the original starting point with a new point which corresponds to y_0 on the original scale (e.g., as the French Revolution did, select 1789 as the new Year 0), then, instead as the original numerical value y, we get a new numerical value $y' = y - y_0$.

In general, if we change both the measuring unit and the starting point, then instead of the original numerical value y, we get the new value $\lambda \cdot y - y_0$.

We should select a family of polynomials. Because of scaling and shift, for each polynomial $f(x)$, the polynomials $\lambda \cdot f(x) - y_0$ represent the same dependence, but expressed in different units. Because of this fact, we should not select a *single* polynomial, we should select the entire *family* $\{\lambda \cdot f(x) - y_0\}_{\lambda, y_0}$ of polynomials representing the original dependence for different selections of the measuring unit and the starting point.

Scaling and shift for input variables. In many practical applications, the inputs numbers x_i are values of some physical quantities. The numerical value of each such quantity also depends on the choice of a measuring unit and on the choice of the starting point. By using different choices, we get new values $x'_i = \lambda_i \cdot x_i - x_{i0}$, for some values λ_i and x_{i0}.

Transformations corresponding to a given endpoint $x^{(0)} = \left(x_1^{(0)},\ldots,x_n^{(0)}\right)$. Once the endpoint is given, we no longer have the freedom of changing the starting point, but we still have re-scalings: $x_i - x_i^{(0)} \to \lambda_i \cdot \left(x_i - x_i^{(0)}\right)$, i.e., equivalently, $x_i \to x'_i = x_i^{(0)} + \lambda \cdot \left(x_i - x_i^{(0)}\right)$.

What is meant by "the best" family? When we say "the best" family, we mean that on the set of all the families, there is a relation $\succeq$ describing which family is better or equal in quality. This relation must be transitive (if $\mathcal{F}$ is better than $\mathcal{G}$, and $\mathcal{G}$ is better than $\mathcal{H}$, then $\mathcal{F}$ is better than $\mathcal{H}$).

Final optimality criteria. The preference relation $\succeq$ is not necessarily asymmetric, because we can have two families of the same quality. However, we would like to require that this relation be *final* in the sense that it should define a unique *best* family $\mathcal{F}_{\rm opt}$, for which $\forall \mathcal{G}\,(\mathcal{F}_{\rm opt} \succeq \mathcal{G})$.

Indeed, if none of the families is the best, then this criterion is of no use, so there should be *at least one* optimal family.

If *several* different families are equally best, then we can use this ambiguity to optimize something else: e.g., if we have two families with the same approximating quality, then we choose the one which is easier to compute. As a result, the original criterion was not final: we obtain a new criterion: $\mathcal{F} \succeq_{\rm new} \mathcal{G}$, if either $\mathcal{F}$ gives a better approximation, or if $\mathcal{F} \sim_{\rm old} \mathcal{G}$ and $\mathcal{G}$ is easier to compute. For the new optimality criterion, the class of optimal families is narrower.

We can repeat this procedure until we obtain a final criterion for which there is only one optimal family.

Optimality criteria should be invariant. Which of the two families is better should not depend on the choice of measuring units for measuring the inputs x_i. Thus, if $\mathcal{F}$ was better than $\mathcal{G}$, then after re-scaling, the re-scaled family $\mathcal{F}$ should still be better than the re-scaled family $\mathcal{G}$.

Thus, we arrive at the following definitions.

3 Optimal Functions Corresponding to Endpoints: Definitions and the Main Result

Definition 1. *By a* family*, we mean a set of functions from $\mathbb{R}^n \to \mathbb{R}$ which has the form $\{C \cdot f(x) - y_0 : C, y_0 \in \mathbb{R}, C > 0\}$ for some polynomial $f(x)$. Let $\mathcal{F}$ denote the class of all possible families.*

Definition 2. *By a* optimality criterion *$\preceq$ on the class $\mathcal{F}$, we mean a pre-ordering relation on the set $\mathcal{F}$, i.e., a transitive relation for which $F \preceq F$ for every F. We say that a family F is optimal with respect to the optimality criterion $\preceq$ if $G \preceq F$ for all $G \in \mathcal{F}$.*

Definition 3. *We say that the optimality criterion is* final *if there exists one and only one optimal family.*

Definition 4. *Let $x^{(0)}$ ba a vector. By a $x^{(0)}$-*rescaling *corresponding to the values $\lambda = (\lambda_1, \ldots, \lambda_n)$, $\lambda_i > 0$, we mean a transformation $x \to x' = T_{x^{(0)},\lambda}(x)$ for which*

$$x'_i = x_i^{(0)} + \lambda_i \cdot \left(x_i - x_i^{(0)}\right).$$

*By a $x^{(0)}$-*rescaling *of a family $F = \{C \cdot f(x) - y_0\}_{C,y_0}$, we mean a family $T_{x^{(0)},\lambda}(F) = \{C \cdot f(T_{x^{(0)},\lambda}(x))s\}_{C,y_0}$. We say that an optimality criterion is $x^{(0)}$-*scaling-invariant *if for every F, G, and λ, $F \preceq G$ implies $T_{x^{(0)},\lambda}(F) \preceq T_{x^{(0)},\lambda}(G)$.*

Proposition 1. *Let $\preceq$ be a final $x^{(0)}$-scaling-invariant optimality criterion. Then every polynomial from the optimal family has the form*

$$f(x) = A + B \cdot \prod_{i=1}^{n} \left(x_i - x_i^{(0)}\right)^{k_i}.$$

Comment. For readers' convenience, all the proofs are placed in the special (last) Proofs section.

Discussion. As we have mentioned, the value of each quantity is defined modulo a starting point. It is therefore reasonable, for y, to select a starting point so that $A = 0$. Thus, we get the dependence

$$f(x) = B \cdot \prod_{i=1}^{n} \left(x_i - x_i^{(0)}\right)^{k_i}.$$

Once the starting point for y is fixed, the only remaining y-transformations are scalings $y \to \lambda \cdot y$.

4 Optimal Combination Operations

In the previous section, we described the optimal functions corresponding to the endpoints $\underline{x}$ and $\overline{x}$. What is the optimal way of combining these functions? Since we are dealing only with polynomial functions, it is reasonable to require that a combination operation transform polynomials into polynomials.

Definition 5. *By a* combination operation*, we mean a function $K : \mathbb{R}^2 \to \mathbb{R}$ for which, if $\underline{f}(x)$ and $\overline{f}(x)$ are polynomials, then the composition $K\left(\underline{f}(x), \overline{f}(x)\right)$ is also a polynomial.*

Lemma 1. *A function $K(a, b)$ is a combination operation if and only if it is a polynomial.*

Discussion. Similarly to the case of optimal functions corresponding to individual endpoint, the numerical value of the function $K\left(\underline{a}, \overline{a}\right)$ depends on the choice of the measuring unit and the starting point: an operation that has the form $K\left(\underline{a}, \overline{a}\right)$ under one choice of the measuring unit and starting point has the form $C \cdot K\left(\underline{a}, \overline{a}\right) - y_0$ under a different choice. Thus, we arrived at the following definition.

Definition 6. *By a* C-family, *we mean a set of functions from* $\mathrm{I\!R}^2 \to \mathrm{I\!R}$ *which has the form* $\{C \cdot K(a,b) - y_0 : C, y_0 \in \mathrm{I\!R}, C > 0\}$ *for some combination operation* $K(a,b)$*. Let* $\mathcal{K}$ *denote the class of all possible C-families.*

Definition 7. *By an* optimality criterion $\preceq$ *on the class* $\mathcal{K}$ *of all C-families, we mean a pre-ordering relation on the set* $\mathcal{K}$*, i.e., a transitive relation for which* $F \preceq F$ *for every C-family* F*. We say that a C-family* F *is optimal with respect to the optimality criterion* $\preceq$ *if* $G \preceq F$ *for all* $G \in \mathcal{K}$*.*

Definition 8. *We say that the optimality criterion is* final *if there exists one and only one optimal C-family.*

Discussion. From the previous section, we know that both functions $\underline{f}(x)$ and $\overline{f}(x)$ are determined modulo scaling $\underline{f}(x) \to \underline{\lambda} \cdot \underline{f}(x)$ and $\overline{f}(x) \to \overline{\lambda} \cdot \overline{f}(x)$. Thus, it is reasonable to require that the optimality relation not change under such re-scalings.

Definition 9. *By a* C-rescaling *corresponding to the values* $\lambda = \left(\underline{\lambda}, \overline{\lambda}\right)$*, we mean a transformation* $T_\lambda\left(\underline{a}, \overline{a}\right) = \left(\underline{\lambda} \cdot \underline{a}, \overline{\lambda} \cdot \overline{a}\right)$*. By a* C-rescaling *of a family*

$$F = \{C \cdot K(\underline{a}, \overline{a}) - y_0\}_{C,y_0},$$

we mean a family $T_\lambda(F) = \{C \cdot K(T_\lambda(a))\}_{C,y_0}$*. We say that an optimality criterion is* C-scaling-invariant *if for every* F*,* G*, and* λ*,* $F \preceq G$ *implies* $T_\lambda(F) \preceq T_\lambda(G)$*.*

Proposition 2. *Let* $\preceq$ *be a final C-scaling-invariant optimality criterion. Then every combination operation from the optimal family has the form*

$$K(\underline{a}, \overline{a}) = A + B \cdot \underline{a}^{\underline{k}} \cdot \overline{a}^{\overline{k}}.$$

5 Conclusions

By applying this optimal combination operation from Section 4 to the optimal functions corresponding to $x^{(0)} = \underline{x}$ and $x^{(0)} = \overline{x}$ (described in Section 3), we conclude that the resulting function has the form

$$f(x_1, \ldots, x_n) = K\left(\underline{f}(x_1, \ldots, x_n), \overline{f}(x_1, \ldots, x_n)\right) =$$

$$A + B \cdot \left(\prod_{i=1}^n (x_i - \underline{x}_i)^{\underline{k}_i}\right)^{\underline{k}} \cdot \left(\prod_{i=1}^n (\overline{x}_i - x_i)^{\overline{k}_i}\right)^{\overline{k}}.$$

Modulo an additive constant, this function has the form

$$f(x_1, \ldots, x_n) = B \cdot \prod_{i=1}^n (x_i - \underline{x}_i)^{\underline{k}'_i} \cdot \prod_{i=1}^n (\overline{x}_i - x_i)^{\overline{k}'_i},$$

where $\underline{k}'_i = \underline{k}_i \cdot \underline{k}$ and $\overline{k}'_i = \overline{k}_i \cdot \overline{k}$.

These are Bernstein polynomials. Thus, Bernstein polynomials can indeed by uniquely determined as the result of applying an optimal combination operation to optimal functions corresponding to $\underline{x}$ and $\overline{x}$.

6 Proofs

Proof of Proposition 1.

1°. Let us first prove that the optimal family $F_{\rm opt}$ is $x^{(0)}$-scaling-invariant, i.e., $T_{x^{(0)},\lambda}(F_{\rm opt})=F_{\rm opt}$.

Since $F_{\rm opt}$ is an optimal family, we have $G\preceq F_{\rm opt}$ for all families G. In particular, for every family G and for every λ, we have $T_{x^{(0)},\lambda^{-1}}(G)\preceq F_{\rm opt}$. Since the optimal criterion is $x^{(0)}$-scaling-invariant, we conclude that

$$T_{x^{(0)},\lambda}\left(T_{x^{(0)},\lambda^{-1}}(G)\right)\preceq T_{x^{(0)},\lambda}(F_{\rm opt}).$$

One can easily check that if we first re-scale the family with the coefficient λ^{-1}, and then with λ, then we get the original family G back. Thus, the above conclusion takes the form $G\preceq T_{x^{(0)},\lambda}(F_{\rm opt})$. This is true for all families G, hence the family $T_{x^{(0)},\lambda}(F_{\rm opt})$ is optimal. Since the optimality criterion is final, there is only one optimal family, so $T_{x^{(0)},\lambda}(F_{\rm opt})=F_{\rm opt}$. The statement is proven.

2°. For simplicity, instead of the original variables x_i, let us consider auxiliary variables $z_i=x_i-x_i^{(0)}$. In terms of these variables, re-scaling takes a simpler form $z_i\to\lambda_i\cdot z_i$. Since $x_i=z_i+x_i^{(0)}$, the dependence $f(x_1,\ldots,x_n)$ take the form

$$g(z_1,\ldots,z_n)=f\left(z_1+x_1^{(0)},\ldots,z_n+x_n^{(0)}\right).$$

Since the function $f(x_1,\ldots,x_n)$ is a polynomial, the new function $g(z_1,\ldots,z_n)$ is a polynomial too.

3°. Let us now use the invariance that we have proved in Part 1 of this proof to find the dependence of the function $f(z)$ on each variable z_i. For that, we will use invariance under transformations that change z_i to $\lambda_i\cdot z_i$ and leave all other coordinates z_j $(j\neq i)$ intact.

Let us fix the values z_j of all the variables except for z_i. Under the above transformation, invariance implies that if $g(z_1,\ldots,z_{i-1},z_i,z_{i+1},\ldots,z_n)$ is a function from the optimal family, then the re-scaled function $g(z_1,\ldots,z_{i-1},\lambda_i\cdot z_i,z_{i+1},\ldots,z_n)$ belongs to the same family, i.e.,

$$g(z_1,\ldots,z_{i-1},\lambda_i\cdot z_i,z_{i+1},\ldots,z_n)=C(\lambda_i)\cdot g(z_1,\ldots,z_{i-1},z_i,z_{i+1},\ldots,z_n)-y_0(\lambda_i)$$

for some values C and y_0 depending on λ_i. Let us denote

$$g_i(z_i)=g(z_1,\ldots,z_{i-1},z_i,z_{i+1},\ldots,z_n).$$

Then, the above condition takes the form

$$g_i(\lambda\cdot z_i)=C(\lambda_i)\cdot g_i(z_i)-y_0(\lambda_i).$$

It is possible that the function $g_i(z_i)$ is a constant. If it is not a constant, this means that there exist values $z_i\neq z_i'$ for which $g_i(z_i)\neq g_i(z_i')$. For these two values, we get

$$g_i(\lambda_i\cdot z_i)=C(\lambda_i)\cdot g_i(z_i)-y_0(\lambda_i);$$

$$g_i(\lambda_i\cdot z_i')=C(\lambda_i)\cdot g_i(z_i')-y_0(\lambda_i).$$

By subtracting these equations, we conclude that

$$g_i(\lambda_i \cdot z_i) - g_i(\lambda_i \cdot z'_i) = C(\lambda_i) \cdot (g_i(z_i) - g_i(z'_i)),$$

hence

$$C(\lambda_i) = \frac{g_i(\lambda_i \cdot z_i) - g_i(\lambda_i \cdot z'_i)}{g_i(z_i) - g_i(z'_i)}.$$

Since the function $g_i(z_i)$ is a polynomial, the right-hand side is a smooth function of λ. Thus, the dependence of $C(\lambda_i)$ on λ_i is differentiable (smooth). Since $y_0(\lambda_i) = C(\lambda_i) \cdot g_i(z_i) - g_i(\lambda_i \cdot z_i)$, and both C and g_i are smooth functions, the dependence $y_0(\lambda_i)$ is also smooth.

Since all three functions C, y_0, and g_i are differentiable, we can differentiate both sides of the equality $g_i(\lambda_i \cdot z_i) = C(\lambda_i) \cdot g_i(z_i) - y_0(\lambda_i)$ by λ_i and take $\lambda_i = 1$. This leads to the formula

$$z_i \cdot \frac{dg_i}{dz_i} = C_1 \cdot g_i(z_i) - y_1,$$

where we denoted $C_1 \stackrel{\rm def}{=} \frac{dC}{d\lambda_i}_{|\lambda_i=1}$ and $y_1 \stackrel{\rm def}{=} \frac{dy_0}{d\lambda_i}_{|\lambda_i=1}$.

By moving all the terms related to g_i to one side and all the terms related to z_i to the other side, we get

$$\frac{dg_i}{C_1 \cdot g_i - y_1} = \frac{dz_i}{z_i}.$$

We will consider two possibilities: $C_1 = 0$ and $C_1 \neq 0$.

3.1°. If $C_1 = 0$, then the above equation takes the form

$$-\frac{1}{y_1} \cdot dg_i = \frac{dz_i}{z_i}.$$

Integrating both sides, we get

$$-\frac{1}{y_1} \cdot g_i = \ln(z_i) + \text{const},$$

thus $g_i = -y_1 \cdot \ln(z_i) + \text{const}$. This contradicts to the fact that the dependence $g_i(z_i)$ is polynomial. Thus, $C_1 \neq 0$.

3.2°. Since $C_1 \neq 0$, we can introduce a new variable $h_i = g_i - \frac{y_1}{C_1}$. For this new variable, we have $dh_i = dg_i$. Hence the above differential equation takes the simplified form

$$\frac{1}{C_1} \cdot \frac{dh_i}{h_i} = \frac{dz_i}{z_i}.$$

Integrating both sides, we get

$$\frac{1}{C_1} \cdot \ln(h_i) = \ln(z_i) + \text{const},$$

hence

$$\ln(h_i) = C_1 \cdot \ln(z_i) + \text{const},$$

and

$$h_i = \text{const} \cdot z_i^{C_1}.$$

Thus,

$$g_i(z_i) = h_i(z_i) + \frac{y_1}{C_1} = \text{const} \cdot z_i^{C_1} + \frac{y_1}{C_1}.$$

Since we know that $g_i(z_i)$ is a polynomial, the power C_1 should be a non-negative integer, so we conclude that

$$g_i(z_i) = A \cdot z_i^{k_i} + B$$

for some values A_i, B_i, and k_i which, on general, depend on all the other values z_j.

4°. Since the function $g(z_1, \ldots, z_n)$ is a polynomial, it is continuous and thus, the value k_i continuously depends on z_j. Since the value k_i is always an integer, it must therefore be constant – otherwise we would have a discontinuous jump from one integer to another. Thus, the integer k_i is the same for all the values z_j.

5°. Let us now use the above dependence on each variable z_i to find the dependence on two variables. Without losing generality, let us consider dependence on the variables z_1 and z_2.

Let us fix the values of all the other variables except for z_1 and z_2, and let us define

$$g_{12}(z_1, z_2) = g(z_1, z_2, z_3, \ldots, z_n).$$

Our general result can be applied both to the dependence on z_1 and to the dependence on z_2. The z_1-dependence means that $g_{12}(z_1, z_2) = A_1(z_2) \cdot z_1^{k_1} + B_1(z_2)$, and the z_1-dependence means that $g_{12}(z_1, z_2) = A_2(z_1) \cdot z_2^{k2} + B_2(z_1)$. Let us consider two possible cases: $k_1 = 0$ and $k_1 \neq 0$.

5.1°. If $k_1 = 0$, this means that $g_{12}(z_1, z_2)$ does not depend on z_1 at all, so both A_2 and B_2 do not depend on z_1, hence we have $g_{12}(z_1, z_1) = A_2 \cdot z_2^{k_2} + B_2$.

5.2°. Let us now consider the case when $k_1 \neq 0$. For $z_1 = 0$, the z_1-dependence means that $g_{12}(0, z_2) = B_1(z_2)$, and the z_2-dependence implies that $B_1(z_2) = g_{12}(0, z_2) = A_2(0) \cdot z_2^{k2} + B_2(0)$.

For $z_1 = 1$, the z_1-dependence means that $g_{12}(1, z_2) = A_1(z_2) + B_1(z_2)$. On the other hand, from the z_2-dependence, we conclude that $A_1(z_2) + B_1(z_2) = g_{12}(1, z_2) = A_2(1) \cdot z_2^{k2} + B_2(1)$. We already know the expression for $B_1(z_2)$, so we conclude that

$$A_1(z_2) = g_{12}(1, z_2) - B_1(z_2) = (A_2(1) - A_2(0)) \cdot z_2^{k_2} + (B_2(1) - B_2(0)).$$

Thus, both $A_1(z_2)$ and $B_1(z_2)$ have the form $a + b \cdot z^{k_2}$, hence we conclude that

$$g_{12}(z_1, z_2) = (a + b \cdot z_2^{k_2}) \cdot z_1^{k_1} + (c + d \cdot z_2^{k_2}) = c + a \cdot z_1^{k_1} + d \cdot z_2^{k_2} + b \cdot z_1^{k_1} \cdot z_2^{k_2}.$$

Previously, we only considered transformations of a single variable, let us now consider a joint transformation $z_1 \to \lambda_1 \cdot z_1$, $z_2 \to \lambda_2 \cdot z_2$. In this case, we get

$$g(\lambda_1 \cdot z_1, \lambda_2 \cdot z_2) = c + a \cdot \lambda_1^{k_1} \cdot z_1^{k_1} + d \cdot \lambda_2^{k_2} \cdot z_2^{k_2} + b \cdot \lambda_1^{k_1} \cdot \lambda_2^{k_2} \cdot z_1^{k_1} \cdot z_2^{k_2}.$$

We want to make sure that

$$g(\lambda_1 \cdot z_1, \lambda_2 \cdot z_2) = C(\lambda_1, \lambda_2) \cdot g(z_1, z_2) - y_0(\lambda_1, \lambda_2),$$

i.e., that

$$c + a \cdot \lambda_1^{k_1} \cdot z_1^{k_1} + d \cdot \lambda_2^{k_2} \cdot z_2^{k_2} + b \cdot \lambda_1^{k_1} \cdot \lambda_2^{k_2} \cdot z_1^{k_1} \cdot z_2^{k_2} =$$
$$C(\lambda_1, \lambda_2) \cdot (c + a \cdot z_1^{k_1} + d \cdot z_2^{k_2} + b \cdot z_1^{k_1} \cdot z_2^{k_2}) - y_0(\lambda_1, \lambda_2).$$

Both sides are polynomials in z_1 and z_2; the polynomials coincide for all possible values z_1 and z_2 if and only if all their coefficients coincide. Thus, we conclude that

$$a \cdot \lambda_1^{k_1} = a \cdot C(\lambda_1, \lambda_2);$$

$$d \cdot \lambda_2^{k_2} = d \cdot C(\lambda_1, \lambda_2);$$

$$c \cdot \lambda_1^{k_1} \cdot \lambda_2^{k_2} = c \cdot C(\lambda_1, \lambda_2).$$

If $a \neq 0$, then by dividing both sides of the a-containing equality by a, we get $C(\lambda_1, \lambda_2) = \lambda_1^{k_1}$. If $d \neq 0$, then by dividing both sides of the d-containing equality by d, we get $C(\lambda_1, \lambda_2) = \lambda_2^{k_2}$. If $c \neq 0$, then by dividing both sides of the c-containing equality by c, we get $C(\lambda_1, \lambda_2) = \lambda_1^{k_1} \cdot \lambda_2^{k_2}$. These three formulas are incompatible, so only one of three coefficients a, d, and c is different from 0 and two other coefficients are equal to 0. In all three cases, the dependence has the form

$$g_{12}(z_1, z_2) = a + \text{const} \cdot z_1^{\ell_1} \cdot z_2^{\ell_2}.$$

6°. Similarly, by considering more variables, we conclude that

$$g(z_1, \ldots, z_n) = a + \text{const} \cdot z_1^{\ell_1} \cdot \ldots \cdot z_n^{\ell_n}.$$

By plugging in the values z_i in terms of x_i, we get the conclusion of the proposition. The proposition is proven.

Proof of Lemma 1. Let us first show that if the function $K(a, b)$ is a combination operation, then $K(a, b)$ is a polynomial. Indeed, by definition of a combination operation, if we take $\underline{f}(x) = x_1$ and $\overline{f}(x) = x_2$, then the function $f(x) = K\left(\underline{f}(x), \overline{f}(x)\right) = K(x_1, x_2)$ is a polynomial.

Vice versa, if $K(x_1, x_2)$ is a polynomial, then for every two polynomials $\underline{f}(x)$ and $\overline{f}(x)$, the composition $f(x) = K\left(\underline{f}(x), \overline{f}(x)\right)$ is also a polynomial. The lemma is proven.

Proof of Proposition 2. Due to Lemma, Proposition 2 follows from Proposition 1 – for the case of two variables.

Acknowledgements

This work was supported in part by the National Science Foundation grants HRD-0734825 and DUE-0926721 and by Grant 1 T36 GM078000-01 from the National Institutes of Health.

References

[1] M. Berz and G. Hoffstätter, "Computation and Application of Taylor Polynomials with Interval Remainder Bounds", *Reliable Computing*, 1998, Vol. 4, pp. 83–97.

[2] M. Berz and K. Makino, "Verified Integration of ODEs and Flows using Differential Algebraic Methods on High-Order Taylor Models", *Reliable Computing*, 1998, Vol. 4, pp. 361–369.

[3] M. Berz, K. Makino and J. Hoefkens, "Verified Integration of Dynamics in the Solar System", *Nonlinear Analysis: Theory, Methods, & Applications*, 2001, Vol. 47, pp. 179–190.

[4] J. Garloff, "The Bernstein algorithm", *Interval Computation*, 1993, Vol. 2, pp. 154–168.

[5] J. Garloff, "The Bernstein expansion and its applications", *Journal of the American Romanian Academy*, 2003, Vol. 25–27, pp. 80–85.

[6] J. Garloff and B. Graf, "Solving strict polynomial inequalities by Bernstein expansion", In N. Munro, editor, *The Use of Symbolic Methods in Control System Analysis and Design*, volume 56 of IEE Contr. Eng., London, 1999, pp. 339–352.

[7] J. Garloff and A. P. Smith, "Solution of systems of polynomial equations by using Bernstein polynomials", In: G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto (eds.), *Symbolic Algebraic Methods and Verification Methods – Theory and Application*, Springer-Verlag, Wien, 2001, pp. 87–97.

[8] J. Hoefkens and M. Berz, "Verification of Invertibility of Complicated Functions over Large Domains", *Reliable Computing*, 2002, Vol. 8, No. 1, pp. 1–16.

[9] R. Lohner, *Einschliessung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*, Ph.D. thesis, Universität Karlsruhe, Karlsruhe, Germany, 1988.

[10] P. S. V. Nataraj and M. Arounassalame, "A new subdivision algorithm for the Bernstein polynomial approach to global optimization", *International Journal of Automation and Computing*, 2007, Vol. 4, pp. 342–352.

[11] A. Neumaier, "Taylor Forms - Use and Limits", *Reliable Computing*, 2002, Vol. 9, pp. 43–79.

[12] N. S. Nedialkov, V. Kreinovich, and S. A. Starks, "Interval Arithmetic, Affine Arithmetic, Taylor Series Methods: Why, What Next?", *Numerical Algorithms*, 2004, Vol. 37, pp. 325–336.

[13] S. Ray and P. S. V. Nataraj, "A New Strategy For Selecting Subdivision Point In The Bernstein Approach To Polynomial Optimization", *Reliable Computing*, 2010, Vol. 14, pp. 117–137.