A New Strategy For Selecting Subdivision Point In The Bernstein Approach To Polynomial Optimization*

Shashwati Ray
Systems and Control Engineering Group
Room 114A, ACRE Building,
Indian Institute of Technology, Bombay, India 400 076
shashwatiray@yahoo.com

P.S.V. Nataraj
Systems and Control Engineering Group
Room 114A, ACRE Building,
Indian Institute of Technology, Bombay, India 400 076
nataraj@sc.iitb.ac.in

Abstract

In the Bernstein approach to polynomial range finding, we propose a new rule for selection of the point where the domain subdivision is to be done. For a given direction of subdivision, instead of subdividing a box at its midpoint (as is done in the existing literature), we propose to subdivide the box at a point where the partial derivative of the polynomial (in the direction of subdivision) equals zero. The location of this point is estimated using the variation diminishing property of the derivative polynomial in the Bernstein form. We then compare the performance of the proposed rule for subdivision point with that of the existing midpoint subdivision rule, on nine polynomial problems of different dimensions - varying from two to eight dimensions. We evaluate both the rules using three different existing subdivision direction selection rules, and find the proposed rule to be overall more efficient in computational time and number of subdivisions.

Keywords: Range computation, Bernstein polynomials, Polynomial optimization, Bezier curve
AMS subject classifications: 90C26

*Submitted: February 16, 2009; Revised: January 1, 2010; Accepted: March 18, 2010.

1 Introduction

Many problems in real world applications can be reduced to the problem of finding bounds for the range of a multivariate polynomial on an $l-$dimensional box-like domain. Knowledge of the range of a multivariate polynomial has several useful applications in systems and control theory as well as in many other quantitative sciences. It is therefore important to find easy and efficient methods for good approximations to this range.

One such method is based on the expansion of a multivariate polynomial into Bernstein polynomials. Range computations using the Bernstein form relies on the simple idea that if a polynomial is written in the Bernstein basis over a box, the range of the polynomial is bounded by the values of the minimum and maximum Bernstein coefficients [1, 4, 5]. The Bernstein approach gives tight inclusions for the polynomial range and sometimes is even capable of giving the exact range on the given domain. The Bernstein polynomial approach [5] has the advantage that it reduces the necessary (function) evaluations which might be costly if the degree of the polynomial is high.

Once the bounds for the range of a multivariable polynomial are computed in the Bernstein approach, these may be improved by subdivision of the domain or by elevating the degree of the Bernstein polynomials. The former is generally more efficient than the latter [5]. The widely used rules for subdivision direction selection include the 'cyclic', the 'derivative' and the 'maximum width' rules [8]. Once the direction of subdivision is selected, the next task is to find an appropriate point (in this direction), where the subdivision is to be done. In the existing literature, the midpoint is usually chosen as the point of subdivision. In the context of the Bernstein range finding approach, points for subdivision other than the midpoint have not been explored. For instance, the idea of selecting the subdivision point where the polynomial derivative is zero, may be more fruitful and could be investigated.

Recently in [11] the authors have dealt with alternate choices for the point of subdivision. According to them, the subdivision point is selected depending on the location of the minimum Bernstein coefficient. Using a direction selection rule, subdivision is performed along this direction at the point where the minimum Bernstein coefficient occurs. When the subdivision point lies on the edge, then no subdivision would be performed in the selected direction. Hence, this procedure would work only with a modification of the existing subdivision direction selection rules. Since we are concerned with the selection of subdivision point using any existing subdivision direction selection rule, we are not considering this method for our study.

Motivated by these findings we aim to develop a more efficient strategy for subdivision, so that overall reductions in the number of subdivisions, and thereby overall reductions in computational time and memory are obtained. With this objective, we propose a new rule for subdivision point selection to be applied to a 'basic' Bernstein polynomial based algorithm for polynomial range finding (see section 2.3). Our idea is based on the heuristic that monotonicity often leads to satisfaction of the vertex condition. So, in the chosen subdivision direction, instead of subdividing a box at its midpoint, we propose to subdivide it at a point close to where the partial derivative of the polynomial becomes equal to zero. The location of this point is estimated using the Bernstein form of the derivative polynomial. By doing so, we expect the polynomial to become thereby monotonic (in this direction), at least over one of the resulting subboxes. This may lead to satisfaction of the vertex condition on the subbox and thereby accelerate the range finding algorithm.

We then compare the performance of the proposed rule for subdivision point with that of the existing midpoint subdivision rule, on nine polynomial problems of different dimensions, using three different existing subdivision direction selection rules.

The rest of the paper is organized as follows. In section 2, we give the notations and definitions of the Bernstein polynomials, the subdivision procedure along with the various existing subdivision direction selection rules and a basic algorithm for polynomial range finding based on the Bernstein approach. In section 3, we describe the proposed rule and also present an algorithm for the same. In section 4, we test and compare the performance of the proposed rule with that of the midpoint rule, on nine test problems of different dimensions with each of the three widely used existing rules for subdivision direction selection. In section 5, we conclude the study.

2 Background

2.1 Bernstein form

Following the notations given in [8], let $l \in \mathbb{N}$ be the number of variables and $x = (x_1, x_2, ..., x_l) \in \mathbb{R}^l$. A multi-index I is defined as $I = (i_1, i_2, ..., i_l) \in \mathbb{N}^l$ and multi-power x^I is defined as $x^I = x_1^{i_1} x_2^{i_2} ... x_l^{i_l}$. A multi-index of maximum degrees N is defined as $N = (n_1, n_2, ..., n_l)$ and associate the index $N_{r,-k} = (n_1, ..., n_{r-1}, n_r - k, n_{r+1}, ..., n_l)$, where $0 \leq n_r - k \leq n_r$. Inequalities $I \leq N$ for multi-indices are meant component-wise, where $0 \leq i_k \leq n_k$, $k = 1, 2, ..., l$. With $I = (i_1, ..., i_{r-1}, i_r, i_{r+1}, ..., i_l)$ we associate the index $I_{r,k}$ given by $I_{r,k} = (i_1, ..., i_{r-1}, i_r + k, i_{r+1}, ..., i_l)$, where $0 \leq i_r + k \leq n_r$. Also, we write $\binom{N}{I}$ for $\binom{n_1}{i_1}, ..., \binom{n_l}{i_l}$ and I/N for $(i_1/n_1, i_2/n_2, ..., i_l/n_l)$.

Let $\mathbf{x} = [\underline{x}, \overline{x}]$, $\overline{x} \geq \underline{x}$ be a real interval, where $\underline{x}$=inf $\mathbf{x}$ is the infimum, and $\overline{x}$=sup $\mathbf{x}$ is the supremum of the interval $\mathbf{x}$. The width of the interval $\mathbf{x}$ is defined as wid $\mathbf{x} = \overline{x} - \underline{x}$. For an l-dimensional interval vector or box $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l)$, the width of $\mathbf{x}$ is wid $\mathbf{x}$ = (wid $\mathbf{x}_1$,wid $\mathbf{x}_2$, ...,wid $\mathbf{x}_l$).

An l-variate polynomial p of degree N is written in the power form as

$$p(x) = \sum_{I \leq N} a_I x^I, a_I \in \mathbb{R}, x = (x_1, x_2, ..., x_l) \in \mathbb{R}^l \tag{1}$$

We can expand the multivariate polynomial in (1) into Bernstein polynomials over an l-dimensional box $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_l})$. Without loss of generality, we consider the unit box $\mathbf{u} = [0, 1]^l$, since any nonempty box $\mathbf{x}$ of $\mathbb{R}^l$ can be affinely mapped onto $\mathbf{u}$.

The transformation of a polynomial from its power form (1) into its Bernstein form results in

$$p(x) = \sum_{I \leq N} b_I(\mathbf{u}) B_{N,I}(x), x \in \mathbf{u} \tag{2}$$

The coefficients $b_I(\mathbf{u})$ are called the Bernstein coefficients of p over $\mathbf{u}$, and $B_{N,I}(x)$ is called the I^{th} Bernstein polynomial of degree N defined as

$$B_{N,I}(x) = B_{i_1}^{n_1}(x_1) B_{i_2}^{n_2}(x_2) ... B_{i_l}^{n_l}(x_l)$$

where,

$$B_{i_j}^{n_j}(x_j) = \binom{n_j}{i_j} x_j^{i_j} (1 - x_j)^{n_j - i_j}, i_j = 0, ..., n_j, j = 1, ..., l$$

Each set of coefficients (a_I or b_I) in (1) and (2) can be computed from the other as [2] :

$$a_I = \sum_{J \leq I} (-1)^{I-J} \binom{N}{I} \binom{I}{J} b_J$$

$$b_I(\mathbf{u}) = \sum_{J \le I} \frac{\binom{I}{J}}{\binom{N}{J}} a_J, I \le N \tag{3}$$

The Bernstein coefficients are collected in an array $B(\mathbf{u}) = (b_I(\mathbf{u}))_{I \in S}$, where $S = \{I : I \le N\}$. This array is called as a *Bernstein patch.*

Let $\overline{p}(\mathbf{u})$ denote the range of the polynomial p on $\mathbf{u}$. Then, by the *range enclosing property* of the Bernstein coefficients [8],

$$\overline{p}(\mathbf{u}) \subseteq [\min B(\mathbf{u})\ ,\ \max B(\mathbf{u})] \tag{4}$$

The enclosure interval on the right is called the *Bernstein range enclosure* and denoted as $\widehat{p}(\mathbf{u})$. Similarly, the *Bernstein range enclosure* on any other box $\mathbf{x}$ is denoted as $\widehat{p}(\mathbf{x})$.

Let S_0 be a special subset of the index set S comprising of indices of the vertices of the array $B(\mathbf{u})$, i.e., let

$$S_0 := \{0, n_1\} \times ... \times \{0, n_l\}$$

Then, the lower (*resp.*, upper) bound of the Bernstein range enclosure (4) is sharp if and only if $\min b_I(\mathbf{u})_{I \in S}$ (*resp.*, $\max b_I(\mathbf{u})_{I \in S}$) is attained at a Bernstein coefficient $b_I(\mathbf{u})$ with $I \in S_0$. This condition is known as the *vertex condition* [5]. The vertex condition holds also for any subbox $\mathbf{d} \subseteq \mathbf{u}$ [9]. Further, the vertex condition is said to be met *within a given tolerance* ε, if

$$\min_{S_0} B(\mathbf{u}) - \min B(\mathbf{u}) \le \varepsilon \text{ and } \max B(\mathbf{u}) - \max_{S_0} B(\mathbf{u}) \le \varepsilon \tag{5}$$

For the univariate case with the domain $[\underline{x}, \overline{x}]$, the control points $\mathbf{b_i}$ associated with the Bernstein coefficient b_i are defined as

$$\mathbf{b_i} = (\underline{x} + \frac{i}{n_1}(\overline{x} - \underline{x}), b_i),\ i = 0, ..., n_1$$

The control points are illustrated in Figure 1 for $n_1 = 5$ and $\mathbf{x} = [0, 1]$. The control points are seen to be evenly spaced on the horizontal axis.

Useful geometric properties can be associated with a polynomial in the Bernstein form, by casting it as a Bezier curve [13]. The *control polygon* (or *polyline*) is formed by the set of line segments connecting adjacent control points [3]. The *convex hull property* [13] states that the Bezier curve lies in the convex hull of its control polygon (as illustrated in Figure 1).

Analogous properties hold for the multivariate cases. For the multivariate case, the control points are $(I/N, b_I(\mathbf{u})) : I \in S$. The *convex hull property* is [6]

$$conv\,\{(x, p(x)) : x \in \mathbf{u}\} \subseteq conv\,\{(I/N, b_I(\mathbf{u})) : I \in S\} \tag{6}$$

where *conv* P denotes the *convex hull* of P, *i.e.* the smallest convex set containing the set P. Thus, the convex hull property states that the range $\overline{p}(\mathbf{u})$ is contained in the convex hull of the control points.

Let $\mathbf{d}$ be a subbox of $\mathbf{u}$. In the Bernstein form, the first partial derivative of the polynomial p in (1) with respect to x_r $(1 \le r \le l)$ is given by [8]

$$\begin{aligned} p_r^{'}(x) &= \frac{\partial p}{\partial x_r}(x) \\ &= n_r \sum_{I \le N_{r,-1}} \left[b_{I_{r,1}}(\mathbf{d}) - b_I(\mathbf{d}) \right] B_{N_{r,-1},I}(x),\ x \in \mathbf{d} \end{aligned} \tag{7}$$

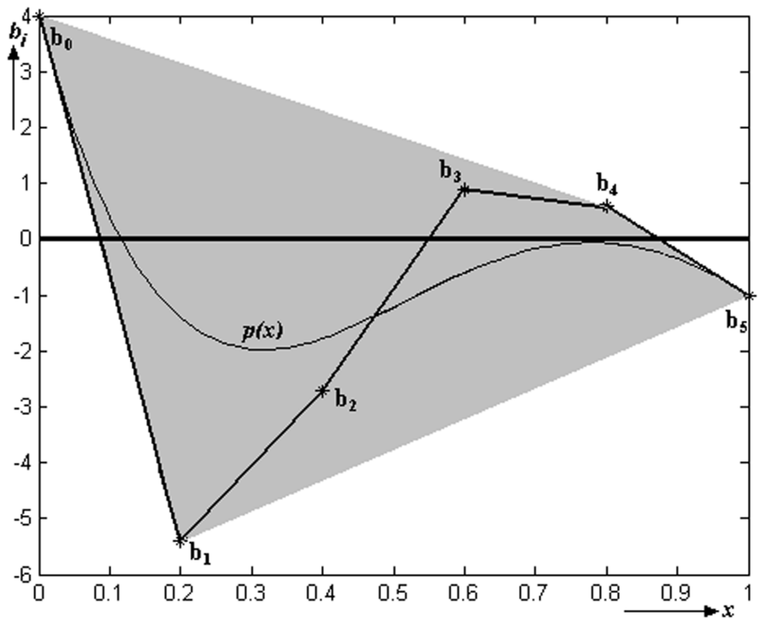

Figure 1: The curve of a univariate polynomial $p(x)$ of fifth degree, with the control points marked by '$*$', the control polygon shown by dark lines, and the convex hull shown as the shaded area.

Thus, the Bernstein coefficients $b_I^{'}$ of the first partial derivative of p with respect to x_r can be obtained simply by forming the differences of its successive Bernstein coefficients. Define

$$b_I^{'}(\mathbf{d}) := n_r \left(b_{I_{r,1}}(\mathbf{d}) - b_I(\mathbf{d})\right) \tag{8}$$

Then,

$$p_r^{'}(x) = \sum_{I \leq N_{r,-1}} b_I^{'}(\mathbf{d}) B_{N_{r,-1},I}(x),\ x \in \mathbf{d} \tag{9}$$

The second partial derivative of p with respect to x_r is given by

$$\begin{aligned} \frac{\partial^2 p}{\partial x_r^2}(x) &= n_r(n_r - 1) \sum_{I \leq N_{r,-2}} \left[b_{I_{r,2}}(\mathbf{d}) - 2b_{I_{r,1}}(\mathbf{d}) + b_I(\mathbf{d})\right. \\ &\quad \left. + b_I(\mathbf{d})\right] B_{N_{r,-2},I}(x),\ \ x \in \mathbf{d} \\ &= (n_r - 1) \sum_{I \leq N_{r,-2}} \left[b_{I_{r,1}}^{'}(\mathbf{d})\right. \\ &\quad \left. - b_I^{'}(\mathbf{d})\right] B_{N_{r,-2},I}(x),\ \ x \in \mathbf{d} \end{aligned} \tag{10}$$

Thus, the Bernstein coefficients $b_I^{''}$ of the second partial derivative of p with respect to x_r can be obtained simply by forming the differences of the successive Bernstein coefficients of the derivative of the polynomial p. Define

$$b_I^{''}(\mathbf{d}) := (n_r - 1) \left(b_{I_{r,1}}^{'}(\mathbf{d}) - b_I^{'}(\mathbf{d})\right) \tag{11}$$

Then,

$$p_r^{''}(x) = \sum_{I \leq N_{r,-1}} b_I^{''}(\mathbf{d}) B_{N_{r,-1},I}(x),\ x \in \mathbf{d} \tag{12}$$

2.2 Subdivision procedure

2.2.1 Subdivision

If we want to tighten the Bernstein range enclosure in (4) when the vertex property does not hold, we may elevate the degree of the Bernstein polynomial [5]. However, a better way to get tighter bounds on the range enclosure is to subdivide the domain box $\mathbf{x}$ into smaller subboxes, and apply the Bernstein expansion to the polynomial p in (1) on the resulting subboxes. Then, the Bernstein range enclosure $\widehat{p}(\mathbf{x})$ is contained in the union of the convex hulls of the control points on the subboxes, see Fig. 2 2).

A subdivision in the r^{th} component direction ($1 \leq r \leq l$) is a subdivision perpendicular to this direction. For simplicity, take the initial domain box to be $\mathbf{u}$, and let

$$\mathbf{d} = [\underline{d}_1, \overline{d}_1] \times ... \times [\underline{d}_r, \overline{d}_r] \times ... \times [\underline{d}_l, \overline{d}_l] \tag{13}$$

be any subbox of $\mathbf{u}$. If $\mathbf{d}$ is subdivided along the r^{th} component direction at some point $\lambda_r \in [0, 1]$, then the resulting two subboxes $\mathbf{d}_A$ and $\mathbf{d}_B$ are

$$\mathbf{d}_A = [\underline{d}_1, \overline{d}_1] \times ... \times [\underline{d}_r, \widehat{d}_r] \times ... \times [\underline{d}_l, \overline{d}_l] \tag{14}$$

$$\mathbf{d}_B = [\underline{d}_1, \overline{d}_1] \times ... \times [\widehat{d}_r, \overline{d}_r] \times ... \times [\underline{d}_l, \overline{d}_l] \tag{15}$$

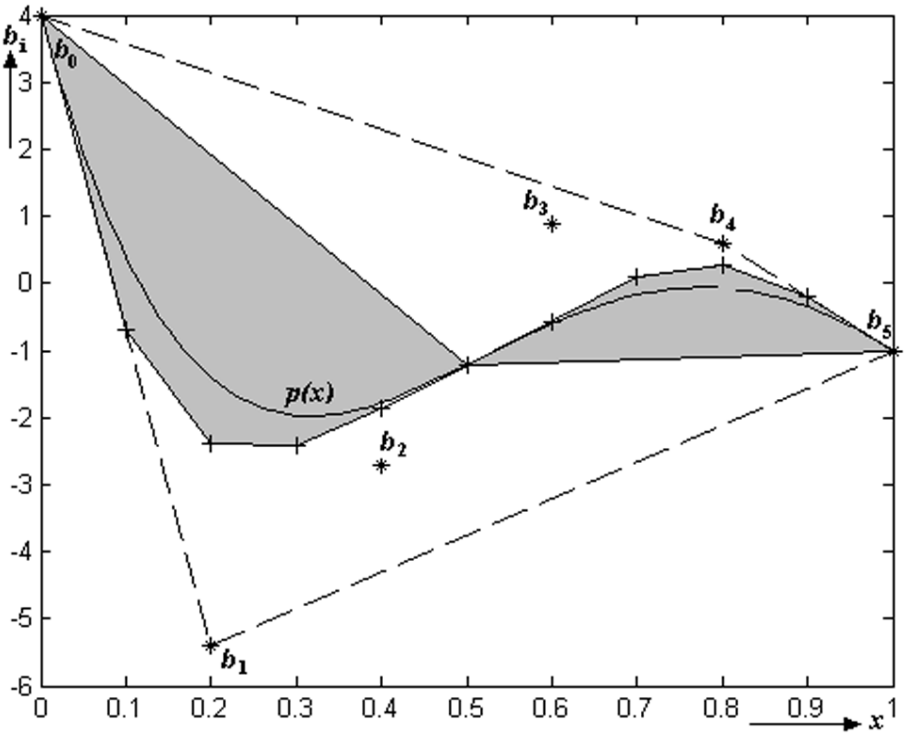

Figure 2: Bounds (shown by broken lines) of a univariate polynomial of fifth degree (Bernstein coefficients are marked by $*$) are improved by subdivision; the resulting convex hulls on the two subboxes (Bernstein coefficients are marked by $+$) are shaded in dark.

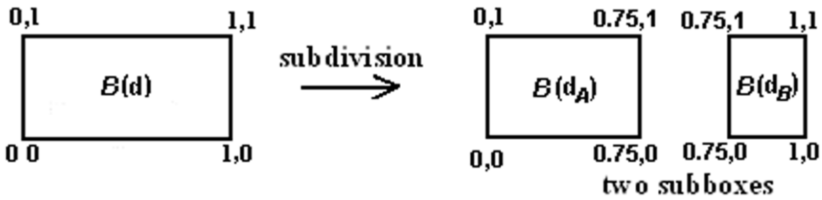

Figure 3: Subdivision in the first coordinate direction with 0.75 as the subdivision point.

where,

$$\widehat{d}_r := \underline{d}_r + \lambda_r(\overline{d}_r - \underline{d}_r) \tag{16}$$

Initially, set $B^0(\mathbf{d}) = B(\mathbf{d})$ for $k = 1, 2, ..., n_r$. Then after subdivision, the Bernstein coefficients can be computed using

$$b^k_{i_1,...,i_r,...,i_l}(\mathbf{d}) = \begin{cases} b^{k-1}_{i_1,...,i_r,...,i_l}(\mathbf{d}) \ : i_r < k \\ (1-\lambda_r) b^{k-1}_{i_1,...,i_r-1,...,i_l}(\mathbf{d}) \\ +\lambda_r b^{k-1}_{i_1,...,i_r,...,i_l}(\mathbf{d}) \ : i_r \geq k \end{cases} \tag{17}$$

The above formula is applied for $i_j = 0, ..., n_j$, $j = 1, ..., r-1, r+1, ..., l$, to obtain the new Bernstein coefficients $B(\mathbf{d}_A)$ and $B(\mathbf{d}_B)$. Now,

$$B(\mathbf{d}_A) = B^{n_r}(\mathbf{d}) \tag{18}$$

The Bernstein coefficients $B(\mathbf{d}_B)$ on the neighboring subbox $\mathbf{d}_B$ are obtained as intermediate values of this computation, since for $k = 0, 1, ..., n_r$, the following relation holds [8]

$$b_{i_1,...,n_r-k,...,i_l}(\mathbf{d}_B) = b^k_{i_1,...,n_r,...,i_l}(\mathbf{d}_A) \tag{19}$$

By this subdivision procedure, the explicit transformation of the subboxes generated by the subdivisions back to $\mathbf{u}$ is avoided. Fig. 3 illustrates the subdivision process in the first coordinate direction for $l = 2$ and $\lambda_1 = 0.75$.

By repeated subdivisions, the Bernstein range enclosure of the given polynomial over a box can be sharpened until they are accurate to the given tolerance.

2.2.2 Rules for direction selection

As mentioned above, the Bernstein range enclosure can be improved by subdividing the domain in a particular direction. The task then is to find an appropriate component direction k $(1 \leq k \leq l)$ for subdivision so that the range is more efficiently obtained. In the context of the Bernstein approach, several rules are available for selection of the subdivision direction [8, 10]. For this purpose, we first define a merit function.

A merit function for subdivision direction selection is [12]:

$$k := \min \left\{ j : j \in \{1, 2, ..., l\} \text{ and } d(j) = \max_{r=1}^{l} d(r) \right\} \tag{20}$$

where, $d(r)$ is determined by the given rule. Thus, if the maximum is achieved in several component directions, the smallest direction is taken for subdivision. Using this merit function, we describe below some of the commonly used direction selection rules.

Rule A (Cyclic) [10] Initially $r = 0$. The subdivision direction is set as

$$\begin{aligned} k := \text{cycle } r \; & (\text{if } r < l, \text{ then replace } r \text{ by } r+1; \\ & \text{if } r = l, \text{ then replace } r \text{ by } r = 1) \end{aligned} \tag{21}$$

Rule B (Derivative based) Garloff and Graf [7] suggest the following :

$$d(r) := \max_{x \in \mathbf{d}} \left| \frac{\partial p}{\partial x_r}(x) \right|$$

where r is the subdivision direction. The first partial derivative of p with respect to x_r is given by (7). The subdivision direction is chosen by estimating

$$\tilde{I}_r := \max_{I \leq N_{r,-1}} n_r \left| b_{I_{r,1}}(\mathbf{d}) - b_I(\mathbf{d}) \right|$$

$$d(r) := \tilde{I}_r \tag{22}$$

Rule C (Maximum width) This is a derivative free rule commonly used in interval analysis [10]. It is based on the interval width, where

$$d(r) := \text{wid } \mathbf{d}_r \tag{23}$$

Then, subdivision is done along the component direction of maximal width.

2.3 Algorithm for range computations

In this section, we describe a 'basic' algorithm based on Bernstein approach for computing the range of a multivariate polynomial. This algorithm is based on the subdivision procedure explained in section 2.2.1, and is on the lines of the depth-first subdivision algorithm given in [5].

In the algorithm, at the outset, we compute the Bernstein coefficients $B(\mathbf{u})$ of the polynomial after transforming the polynomial onto the unit box domain $\mathbf{u}$. We next initialize a list $\mathcal{L}$ with an item ($\mathbf{u}$,$B(\mathbf{u})$), with the domain box $\mathbf{u}$ and the Bernstein patch $B(\mathbf{u})$. We also initialize a solution list $\mathcal{L}^{sol}$ to the empty list. From the list $\mathcal{L}$, we then pick each item ($\mathbf{d}$,$B(\mathbf{d})$) and check for vertex condition satisfaction. If the vertex condition is satisfied within the specified tolerance ε (cf. (5)), then we remove the item from $\mathcal{L}$ and deposit it in the list $\mathcal{L}^{sol}$ as a *solution box*[1], else we retain it in the list $\mathcal{L}$. Following this, we compute the *current range estimate* $\widehat{p}$ as the minimum

[1] Thus, a *solution box* is a box for which the vertex condition is satisfied within the specified tolerance ε, as per. (5).

to maximum of all the Bernstein patches in $\mathcal{L}^{sol}$. All the items retained in $\mathcal{L}$ then undergo a so called *cut-off* test; the *cut-off* test (explained in section 2.4) essentially deletes items in $\mathcal{L}$ which would not contribute to updating the current range estimate $\widehat{p}$. Then, we pick each item from $\mathcal{L}$, delete its entry from $\mathcal{L}$, choose a subdivision direction k, and subdivide the box $\mathbf{d}$ at the midpoint in the direction k into subboxes $\mathbf{d}_A$ and $\mathbf{d}_B$. Subsequently, we compute the Bernstein patches $B(\mathbf{d}_A)$ and $B(\mathbf{d}_B)$ for the resulting subboxes, making use of (17) to (19). We form the corresponding new items $(\mathbf{d}_A,B(\mathbf{d}_A))$ and $(\mathbf{d}_B,B(\mathbf{d}_B))$ and add them to the list $\mathcal{L}$. We continue the entire process till $\mathcal{L}$ becomes empty. At this point, we output the *current range estimate* $\widehat{p}$ as the desired range enclosure, and terminate the algorithm.

We next present the 'basic' algorithm to compute the range of a multivariate polynomial p on a general box domain $\mathbf{x}$.

Algorithm Range : $\widehat{p}(\mathbf{x})$ = Range $(N, a_I, \mathbf{x}, \varepsilon)$

Inputs : Degree N of the polynomial, the coefficient matrix a_I of the polynomial, the l-dimensional domain box, and the tolerance ε to which the range enclosure is to be computed.

Outputs : A range enclosure $\widehat{p}(\mathbf{x})$ computed to the specified tolerance ε.

BEGIN algorithm

1. Compute the Bernstein coefficients $B(\mathbf{u})$ using Garloff's method [5].
2. {Initialize lists}

   $\mathcal{L} \leftarrow \{(\mathbf{u},B(\mathbf{u}))\}$, $\mathcal{L}^{sol} \leftarrow \{\}$.
3. {Start a new iteration}

   If $\mathcal{L}$ is empty go to step 9.
4. {Check, for each box, if the vertex condition is met within ε}

   For each item $(\mathbf{d},B(\mathbf{d}))$ in $\mathcal{L}$, do the following : if $(\mathbf{d},B(\mathbf{d}))$ satisfies the *vertex condition within* ε as given by (5), then enter the item in list $\mathcal{L}^{sol}$, and delete the item entry from $\mathcal{L}$.
5. {Compute the current range estimate}

   Compute $\widehat{p}$ as the minimum to maximum over the second entries of all the items present in $\mathcal{L}^{sol}$ (i.e., over all the Bernstein patches present in $\mathcal{L}^{sol}$).
6. {Prune list $\mathcal{L}$ using the *cut-off* test, see section 2.4}

   $\mathcal{L}$ = Cut_off_test $(\mathcal{L}, \widehat{p})$.
7. {Subdivide and find new Bernstein patches}

   (i) For $i = 1$ to length$(\mathcal{L})$ do

      (a) Pick the i^{th} item $(\mathbf{d},B(\mathbf{d}))$, from list $\mathcal{L}$ and delete its entry from $\mathcal{L}$.

      (b) Using a rule for selection of subdivision direction, choose a component direction k for subdivision of $\mathbf{d}$.

      (c) Subdivide $\mathbf{d}$ at its midpoint in component direction k to generate two subboxes $\mathbf{d}_A$ and $\mathbf{d}_B$ such that $\mathbf{d} = \mathbf{d}_A \cup \mathbf{d}_B$.

      (d) Compute $B(\mathbf{d}_A)$ and $B(\mathbf{d}_B)$ using (17) to (19) and enter the new items $(\mathbf{d}_A, B(\mathbf{d}_A))$ and $(\mathbf{d}_B, B(\mathbf{d}_B))$ at the end of list $\mathcal{L}$.

   (ii) Go to step 3.

8. {Compute the polynomial range enclosure $\widehat{p}(\mathbf{x})$}
   $\widehat{p}(\mathbf{x}) = \widehat{p}$.
9. {Return}
   Return $\widehat{p}(\mathbf{x})$.

END Algorithm

2.4 The cut-off test

The *cut-off* test is performed in step 6 of Algorithm Range. In the previous step 5 of Algorithm Range, $\widehat{p} = [\inf \widehat{p}, \sup \widehat{p}]$ is computed as the current range estimate. Clearly, since

$$\widehat{p} \subseteq \overline{p}(\mathbf{x}) \Rightarrow \inf \overline{p}(\mathbf{x}) \leq \inf \widehat{p} \leq \sup \widehat{p} \leq \sup \overline{p}(\mathbf{x})$$

we can discard from $\mathcal{L}$ all those items $(\mathbf{d}, B(\mathbf{d}))$, for which

$$\inf \overline{p}(\mathbf{x}) \leq \inf \widehat{p} \leq \min B(\mathbf{d}) \text{ and } \max B(\mathbf{d}) \leq \sup \widehat{p} \leq \sup \overline{p}(\mathbf{x})$$

as these do not lead to improvements in the current range estimate $\widehat{p}$.

An algorithm for performing the *cut-off* test in step 6 of Algorithm Range can be given as follows :

Algorithm Cut_off_test : $\mathcal{L}$ = Cut_off_test ($\mathcal{L}$, $\widehat{p}$)
Inputs : Current range estimate $\widehat{p}$, the list $\mathcal{L}$.
Outputs : A pruned list $\mathcal{L}$.
BEGIN Algorithm

1. {Execute for all boxes in the list $\mathcal{L}$}
   For each item $(\mathbf{d}, B(\mathbf{d}))$ in $\mathcal{L}$, do the following : if

   $$\inf \widehat{p} \leq \min B(\mathbf{d}) \text{ and } \sup \widehat{p} \geq \max B(\mathbf{d})$$

   then discard the item $(\mathbf{d}, B(\mathbf{d}))$ from $\mathcal{L}$.
2. {Return}
   Output the pruned list $\mathcal{L}$.

END Algorithm

3 The proposed rule

For simplicity, we first consider the univariate case to describe the idea behind the proposed rule. The derivative of a univariate polynomial $p(x)$ of degree n, gives a polynomial $p^{'}(x)$ of degree $n-1$. The points at which $p^{'}(x)$ becomes zero are the stationary points of $p(x)$. If we subdivide the domain at all the stationary points, the polynomial becomes monotonic over the resulting subdomains. Although monotonicity does not necessarily imply satisfaction of the vertex condition described in section 2.1, as pointed out in [14], in several instances the implication is found to hold true (recall that the satisfaction of the vertex condition on a subdomain implies that the range of $p(x)$ has been found on that subdomain, see section 2.1). Therefore, we attempt to select a stationary point as the point of subdivision. This is the main idea behind the proposed rule.

The derivative $p^{'}(x)$ can be easily calculated from the Bernstein expansion of the polynomial p as (cf. (9))

$$p^{'}(x) = \sum_{i=0}^{n-1} b_i^{'}\mathbf{u})B_i^{n-1}(x),\ x \in \mathbf{u} \tag{24}$$

where, the Bernstein coefficients $b_i^{'}$ of $p^{'}(x)$ are obtained simply by forming the *differences* of its Bernstein coefficients as

$$b_i^{'}(\mathbf{u}) = n\,(b_{i+1}(\mathbf{u}) - b_i(\mathbf{u})) \tag{25}$$

The control points of the derivative polynomial $p^{'}(x)$ are

$$\begin{pmatrix} i/(n-1) \\ b_i^{'}(\mathbf{u}) \end{pmatrix} : i = 0, ..., n-1$$

as illustrated in Figure 4. So, if all the Bernstein coefficients of the *derivative* polynomial have the same sign, then the control polygon does not intersect the abscissa. This implies that there are no roots of $p^{'}(x)$ on $\mathbf{u}$, so the polynomial $p(x)$ is monotonic on $\mathbf{u}$.

By expressing the derivative polynomial $p^{'}(x)$ in the Bernstein form, we can take advantage of its various properties (the geometric insights based on the control polygon) to locate its roots. By the variation diminishing property in section 2.1, we know that the Bezier curve is no more complicated than its control polygon. Moreover, we may observe that the zeros of $p^{'}(x)$, would be somewhat close to the points where the line segments of the control polygon of $p^{'}(x)$ cross the abscissa (see for instance, Figure 4). These latter points can be easily found out, and used as the points of subdivision. The obtained subdivision points are often near the stationary points of $p^{'}(x)$. If we subdivide the domain box at such points, the polynomial may become monotonic over few of the resulting subboxes. But, if we subdivide the domain box at midpoint, there is no such guarantee that the polynomial would be monotonic over any of the resulting subboxes. As a special case, any polynomial having only one stationary point would be monotonic over at least one subbox, when the domain box is subdivided at the midpoint.

In case we have several line segments of the control polygon of $p'(x)$ intersecting the abscissa, we choose the one with maximum absolute slope (steepest ascent or descent). We then take its intersection point with the abscissa as the point of subdivision λ. This point would be in the area around a stationary point, where the variation of the derived curve is most rapid. It could be close to the extrema, so a better current range estimate could be obtained. The required slopes may be computed by forming the differences of the successive Bernstein coefficients of $p^{'}(x)$, i.e., in terms of the Bernstein coefficients $b_i^{''}$ of the second derivative of $p(x)$ given by (11) as

$$b_i^{''}(\mathbf{u}) = (n-1)\left(b_{i+1}^{'}(\mathbf{u}) - b_i^{'}(\mathbf{u})\right) \tag{26}$$

We illustrate the idea with the help of an univariate example having two stationary points.

Example 1 *The univariate polynomial* $p(x) = 2 + 8x - 17x^2 + 10x^3$, $x \in [0, 1]$ *has two stationary points at* $x_a = 0.3333$ *and* $x_b = 0.80$. *Figure 4 shows the plots of this polynomial and its derivative polynomial* $p^{'}(x)$ *along with their respective control points.*

The control polygon of $p^{'}(x)$ *crosses the abscissa at* 0.2353 *and* 0.8461. *Hence, these are two candidate subdivision points,* $\lambda_a = 0.2353$ *and* $\lambda_b = 0.8461$. *At* λ_a*, the control polygon of the derivative polynomial has a slope* $= (b_1^{'} - b_0^{'})/0.5 = b_0^{''}/\,(0.5(n-1)) = -8.5$. *Whereas, at* λ_b *its slope is* $(b_2^{'} - b_1^{'})/0.5 = b_1^{''}/\,(0.5(n-1)) = 6.5$. *Since the slope is steeper at the point* λ_a*, we choose the subdivision point* λ *as* 0.2353. *With* 0.2353 *as the subdivision point, we get vertex condition satisfaction on the box* $[0, 0.2353]$*, hence this box need not be processed further. We obtain the current range estimate as* $[2, 3.07145]$. *With midpoint as the subdivision point, we do not get vertex condition satisfaction, so both* $[0, 0.5]$ *and* $[0.5, 1]$ *need to be further subdivided.*

On similar lines, we can now extend the idea to the multivariate case. Let $\mathbf{d} \in \mathbf{u}$ be the subbox to be subdivided. The first partial derivative of the polynomial $p(x)$ in (1) with respect to x_r of degree N would be a polynomial $p_r^{'}(x)$ of degree $N_{r,-1}$. The derivative polynomial $p_r^{'}(x)$ of the polynomial p in (1) with respect to x_r $(1 \leq r \leq l)$ is given by (7) and (9). The Bernstein coefficients $b_I^{'}$ is given by (8) and repeated here for convenience

$$b_I^{'}(\mathbf{d}) = n_r \left(b_{I_{r,1}}(\mathbf{d}) - b_I(\mathbf{d})\right) \tag{27}$$

The above Bernstein coefficients of the derivative curve form an $N_{r,-1}$ dimensional array. The corresponding control points are

$$\begin{pmatrix} I/(N_{r,-1}) \\ b_I^{'}(\mathbf{d}) \end{pmatrix} : I = 0, ..., N_{r,-1}$$

A point at which $p_r^{'}(x) = 0$ would be 'approximately near' the points where the control polygon of $p_r^{'}(x)$ intersects the abscissa. Among these points, a suitable one based on the below proposed rule is chosen as the point of subdivision λ_r for subdivision of the box $\mathbf{d}$ in the r^{th} direction.

Let S_c be the set of all pairs of control points where the line segments of the control polygon of the derivative curve intersect the abscissa, i.e.,

$$S_c := \left\{ \left\{ \begin{pmatrix} I/(N_{r,-1}) \\ b_I^{'}(\mathbf{d}) \end{pmatrix}, \begin{pmatrix} I_{r,1}/(N_{r,-1}) \\ b_{I_{r,1}}^{'}(\mathbf{d}) \end{pmatrix} \right\} : b_I^{'}(\mathbf{d}) b_{I_{r,1}}^{'}(\mathbf{d}) < 0 \right\} \tag{28}$$

In Example 1, set S_c is given by

$$\begin{aligned} S_c &= \left\{ \left\{ \begin{pmatrix} 0/2 \\ b_0^{'} \end{pmatrix}, \begin{pmatrix} 1/2 \\ b_1^{'} \end{pmatrix} \right\}, \left\{ \begin{pmatrix} 1/2 \\ b_1^{'} \end{pmatrix}, \begin{pmatrix} 2/2 \\ b_2^{'} \end{pmatrix} \right\} \right\} \\ &= \left\{ \left\{ \begin{pmatrix} 0/2 \\ 8 \end{pmatrix}, \begin{pmatrix} 1/2 \\ -9 \end{pmatrix} \right\}, \left\{ \begin{pmatrix} 1/2 \\ -9 \end{pmatrix}, \begin{pmatrix} 2/2 \\ 4 \end{pmatrix} \right\} \right\} \end{aligned} \tag{29}$$

The Bernstein coefficients $b_I^{''}$ of $p^{''}(x)$ is given by (11) and repeated here for convenience.

$$b_I^{''}(\mathbf{d}) = (n_r - 1) \left(b_{I_{r,1}}^{'}(\mathbf{d}) - b_I^{'}(\mathbf{d})\right)$$

For every pair of control points in S_c in (28), we compute $b_I^{''}(\mathbf{d})$. We then choose that pair of control points for which $\left|b_I^{''}(\mathbf{d})\right|$ is the largest, and compute the point of subdivision using the data for this pair as

$$\lambda_r = \frac{\frac{I}{N_{r,-1}} b_{I_{r,1}}^{'}(\mathbf{d}) - \frac{I_{r,1}}{N_{r,-1}} b_I^{'}(\mathbf{d})}{b_{I_{r,1}}^{'}(\mathbf{d}) - b_I^{'}(\mathbf{d})}, \ \lambda_r \in [0, 1] \tag{30}$$

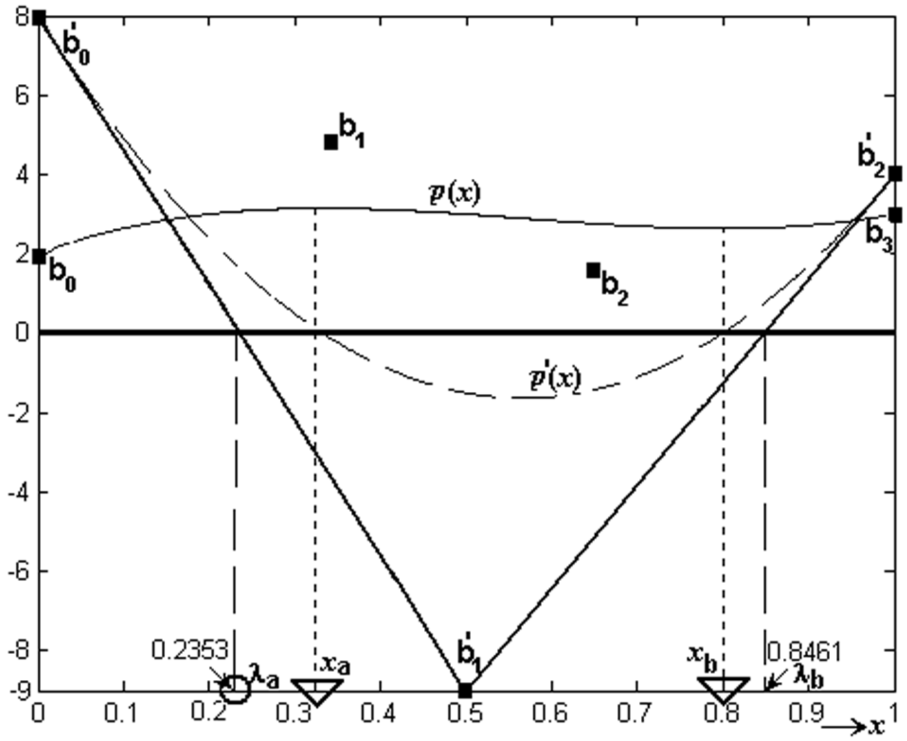

Figure 4: The polynomial $p(x)$ and its derivative $p'(x)$ with respective control points (marked by ■). The stationary point of $p'(x)$ is marked by $\triangledown$; whereas the obtained point of subdivision is marked by ◯.

For the same example, from the set S_c we choose the pair $\left\{ \begin{pmatrix} 0/2 \\ 8 \end{pmatrix}, \begin{pmatrix} 1/2 \\ -9 \end{pmatrix} \right\}$, since $8.5 > 6.5$. Substituting the values in (30) we obtain

$$\lambda_r = \frac{\frac{0}{2}(-9) - \frac{1}{2}8}{-9-8} = 0.2353$$

as the subdivision point, as already reported in Example 1.

Based on the above ideas, we can have an algorithm for selecting the subdivision point.

Algorithm Subdivision_point_selection : λ_r = Subdivision_point_selection **(**$B(\mathbf{d})$, r, N_r**)**

Inputs : Bernstein coefficients $B(\mathbf{d})$ of the box $\mathbf{d}$ to be subdivided, the selected subdivision direction r, and the multi-index of maximum degrees N of each variable of the polynomial p.

Outputs : Subdivision parameter λ_r in the r^{th} component direction.

BEGIN Algorithm

1. {Compute Bernstein coefficients $b_I'(\mathbf{d})$ of $p_r'(x)$}
   $b_I'(\mathbf{d}) = n_r \left(b_{I_{r,1}}(\mathbf{d}) - b_I(\mathbf{d})\right)$.
2. {Compute differences of the successive Bernstein coefficients $b_I'(\mathbf{d})$ wherever the control polygon of the derivative polynomial changes sign}
   Set $\lambda_r := 0.5$
   for $I = 0$ to $N_{r,-1}$ do
   1. if $b_I'(\mathbf{d}) b_{I_{r,1}}'(\mathbf{d}) < 0$ then
      1. Form set S_c using (28)
      2. {Check from S_c}

      If $\left|b_I'(\mathbf{d})\right| \neq \left|b_{I_{r,1}}'(\mathbf{d})\right|$ then compute $b_I''(\mathbf{d}) = (n_r - 1)\left(b_{I_{r,1}}'(\mathbf{d}) - b_I'(\mathbf{d})\right)$
3. {From S_c find the location of maximum value of $b_I''(\mathbf{d})$}
   Choose that element from S_c for which $\left|b_I''(\mathbf{d})\right|$ is maximum.
4. {Use this element from S_c to compute the subdivision point in direction r }
   Compute λ_r using (30)
5. {Return}
   return λ_r.

END Algorithm.

Remark 1 *Subdivision of the box* $\mathbf{d}$ *at* $\widehat{d_r}$ *results in two subboxes* $\mathbf{d}_A$ *and* $\mathbf{d}_B$*. The Bernstein coefficients* $B(\mathbf{d}_B)$ *on the neighboring subbox* $\mathbf{d}_B$ *are obtained as intermediate values of the computation of* $B(\mathbf{d}_A)$*, cf. (17) to (19).*

4 Results and discussion

We test and compare the performance of the proposed rule for subdivision point selection with that of the existing midpoint rule on nine polynomial problems. In the tests, the comparison is done for each basic rule of subdivision direction selection, namely, cyclic, derivative-based, and maximum width rules, i.e., rule A (refer equation 21), rule B (refer equation 22), and rule C (refer equation 23). All the codes are developed in Forte FORTRAN 95 [15], and all computations are performed on a Sun 440 MHz Ultra Sparc 10 Workstation with 2 GB RAM. All rounding errors are accounted for by using interval arithmetic support provided in the compiler.

The application problems are taken from [16, 17, 18] and listed in the Appendix. We use Algorithm Range given in section 2.3 by computing the range enclosures on given domains, to the specified tolerance ε on nine polynomial problems. All the numerical results are obtained with $\varepsilon = 10^{-15}$ for problems of dimension lesser than 7, and with $\varepsilon = 10^{-10}$ for polynomials in higher dimensions.

The performance of the proposed rule for selection of the subdivision point is compared with that of the midpoint rule, in terms of the number of subdivisions required to get the range to the specified accuracy ε, as well as in terms of the computation time required (in seconds) to achieve the same. For the sake of comparison, for each of these metrics we also report the values of the *ratio* computed as

$$\frac{\text{Performance metric with the midpoint rule}}{\text{Performance metric with proposed rule}} = \frac{\text{PMMR}}{\text{PMPR}}$$

and values of the *percent reduction* (% red) computed as

$$\frac{\text{PMMR} - \text{PMPR}}{\text{PMMR}} \times 100$$

Table 1 gives a comparison of the computation time required by both the subdivision point selection rules, while Table 2 gives a comparison of the number of subdivisions required.

From the tables, we see that with the proposed rule for subdivision point selection we are able to solve all the problems, whereas with the midpoint rule we are unable to solve ‘Butchers’ (But 6) and ‘Magnetism’ (Mag 7) problems. Moreover, we observe that with the proposed rule, we are able to compute the ranges of all the test functions in considerably less time and subdivisions. More specifically, with the proposed rule, we are able to get a reduction in computational time varying from 12.95% to almost 100% over the midpoint rule, with an average reduction of 62.65%. There is also a significant reduction in the number of subdivisions; this is varying from 4.56% to 99.95% with an average reduction of 76.11%. Thus, the proposed rule for the subdivision point selection is seen to be considerably more efficient compared to the midpoint rule, in terms of both these performance metrics.

5 Conclusions

We addressed the Bernstein polynomial based approach to polynomial range finding over a given domain. For a given domain box and a given direction of subdivision, we presented a new rule to select the point at which the box is to be subdivided. This subdivision point is located where the first partial derivative of the polynomial with respect to the given subdivision direction becomes zero. The new rule is also capable of

Table 1: Computation time (in seconds) taken by the two rules for subdivision point selection with different subdivision direction selection rules

Prob	dim	time (secs)	Rule A Subdiv point as		Rule B Subdiv point as		Rule C Subdiv point as	
			mid	prop	mid	prop	mid	prop
Quad	2	Number	0.041	0.031	0.041	0.031	0.041	0.031
		Ratio		1.32		1.32		1.32
		% red		24.24		24.24		24.24
Camel	2	number	0.063	0.042	0.078	0.068	0.062	0.052
		Ratio		1.482		1.149		1.191
		% red		32.53		12.95		16.00
R. D. 3	3	Number	0.013	0.004	0.023	0.004	0.013	0.005
		Ratio		3.197		5.615		2.473
		% red		68.72		82.19		59.56
Cap 4	4	Number	0.156	0.109	0.609	0.312	0.203	0.125
		Ratio		1.429		1.950		1.650
		% red		30.00		48.72		38.46
Wrig 5	5	Number	0.047	0.031	0.062	0.016	0.062	0.031
		Ratio		1.500		4.000		2.00
		% red		33.33		75.00		50.00
But 6	6	Number	$>$900	0.031	1.531	0.422	$>$900	0.039
		Ratio		$>$3.2e4		3.630		$>$2.6e4
		% red		$\approx$100		93.77		$\approx$100
Mag 6	6	Number	7.657	0.657	13.609	1.125	7.922	0.422
		Ratio		11.67		12.10		18.78
		% red		91.43		91.73		94.67
Mag 7	7	Number	$>$900	0.547	$>$900	0.687	$>$900	0.531
		Ratio		$>$1.8e3		$>$1.5e3		$>$1.9e3
		% red		$>$99.94		$>$99.93		$>$99.95
Heart 8	8	Number	913.85	45.08	6.055	1.817	5.306	3.451
		Ratio		20.27		3.332		1.537
		% red		95.07		69.99		34.95

deciding (depending on the second partial derivative), whether the new selected point or the midpoint is the appropriate splitting point. Through nine multidimensional examples, we show that the proposed rule yields considerable reductions over the widely used midpoint rule, both in computational time and in the number of subdivisions, for computing polynomial ranges.

Table 2: Number of subdivisions required by the two rules for subdivision point selection with different subdivision direction selection rules

Prob	dim	Sub-division	Rule A		Rule B		Rule C	
			Subdiv point as		Subdiv point as		Subdiv point as	
			mid	prop	midpoint	prop	midpoint	prop
Quad	2	Number	231	7	240	8	231	38
		Ratio		33		30		23.10
		% red		96.97		96.67		95.57
Camel	2	Number	379	345	373	356	379	366
		Ratio		1.10		1.05		1.06
		% red		8.97		4.56		6.07
R. D. 3	3	Number	80	6	120	2	80	7
		Ratio		13.33		60		11.43
		% red		92.50		98.33		91.25
Cap 4	4	Number	823	457	1363	682	823	424
		Ratio		1.80		2.00		1.94
		% red		44.47		49.96		48.48
Wrig 5	5	Number	161	36	223	2	161	32
		Ratio		4.47		111.5		5.03
		% red		77.64		99.10		80.12
But 6	6	Number	>2e04	11	909	192	>2e04	15
		Ratio		>1.9e3		4.73		>1.4e3
		% red		>99.95		78.88		>99.93
Mag 6	6	Number	5247	447	5247	447	5247	287
		Ratio		11.74		11.74		18.28
		% red		91.48		91.48		94.53
Mag 7	7	Number	>5111	127	>5111	127	>5111	127
		Ratio		>40.24		40.24		40.24
		% red		>97.51		>97.51		>97.51
Heart 8	8	Number	2227	342	210	43	208	101
		Ratio		6.512		4.88		2.059
		% red		84.64		79.52		51.44

Acknowledgements

The authors thank Prof. Jürgen Garloff of University of Applied Sciences in Constance, Germany for his comments and suggestions on the work reported in this paper.

Appendix

Description of test problems

In the following, we list the polynomials p, the domain boxes $\mathbf{x}$, the abbreviated and full names, and the dimensionality of the problems used in our tests. The problems are arranged in the order of increasing dimensionality. Except for the first two problems all the test problems are from Verschelde's PHC pack [16].

1. **Quad 2** : Quadratic function $l = 2$ [17]
$$p(x) = x_1^2 + x_2^2 - 2$$
$$\mathbf{x}_1 = [-99.99, 100], \mathbf{x}_2 = [-99.99, 100]$$
2. **Camel 2** : Six hump back camel function $l = 2$ [18]
$$p(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$
$$\mathbf{x}_1 = [-3, 3], \mathbf{x}_2 = [-3, 3]$$
3. **R. D. 3** : A 3-dimensional reaction diffusion problem, $l = 3$
$$p(x_1, x_2, x_3) = x_1 - 2x_2 + x_3 + .835634534x_2(1 - x_2)$$
$$\mathbf{x_1} = [-5, 5],\ \mathbf{x_2} = [-5, 5],\ \mathbf{x_3} = [-5, 5]$$
4. **Cap 4** : Caprasse's system **:** $l = 4$
$$\begin{aligned} p(x_1, x_2, x_3, x_4) &= -x_1x_3^3 + 4x_2x_3^2x_4 + 4x_1x_3x_4^2 + 2x_2x_4^3 \\ &\quad +4x_1x_3 + 4x_3^2 - 10x_2x_4 - 10x_4^2 + 2 \end{aligned}$$
$$\mathbf{x_1} = [-.5, .5],\ \mathbf{x_2} = [-.5, .5],\ \mathbf{x_3} = [-.5, .5],\ \mathbf{x_4} = [-.5, .5]$$
5. **Wrig 5** : System of A.H. Wright, $l = 5$
$$p(x_1, x_2, x_3, x_4, x_5) = x_5^2 + x_1 + x_2 + x_3 + x_4 - x_5 - 10$$
$$\begin{aligned} \mathbf{x_1} &= [-5, 5],\ \mathbf{x_2} = [-5, 5],\ \mathbf{x_3} = [-5, 5],\ \mathbf{x_4} = [-5, 5], \\ \mathbf{x_5} &= [-5, 5] \end{aligned}$$
6. **But 6** : Butcher's problem, $l = 6$
$$p(x_1, x_2, x_3, x_4, x_5, x_6) = x_6x_2^2 + x_5x_3^2 - x_1x_4^2 + x_4^3 + x_4^2 - 1/3x_1 + 4/3x_4$$
$$\begin{aligned} \mathbf{x_1} &= [-1, 0],\ \mathbf{x_2} = [-.1, .9],\ \mathbf{x_3} = [-.1, .5], \mathbf{x_4} = [-1, -.1], \\ \mathbf{x_5} &= [-.1, -.05],\ \mathbf{x_6} = [-.1, -.03] \end{aligned}$$
7. **Mag 6** : A problem of magnetism in physics, $l = 6$
$$p(x_1, x_2, x_3, x_4, x_5, x_6) = 2x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2 + x_6^2 - x_6$$
$$\begin{aligned} \mathbf{x_1} &= [-5, 5],\ \mathbf{x_2} = [-5, 5],\ \mathbf{x_3} = [-5, 5], \\ \mathbf{x_4} &= [-5, 5],\ \mathbf{x_5} = [-5, 5],\ \mathbf{x_6} = [-5, 5] \end{aligned}$$
8. **Mag 7** : Katsura 6, a problem of magnetism in physics, $l = 7$
$$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = x_1^2 + 2x_2^2 + 2x_3^2 + 2x_4^2 + 2x_5^2 + 2x_6^2 + 2x_7^2 - x_1$$
$$\begin{aligned} \mathbf{x_1} &= [-5, 5],\ \mathbf{x_2} = [-5, 5],\ \mathbf{x_3} = [-5, 5],\ \mathbf{x_4} = [-5, 5], \\ \mathbf{x_5} &= [-5, 5],\ \mathbf{x_6} = [-5, 5],\ \mathbf{x_7} = [-5, 5] \end{aligned}$$

9. **Heart 8** : Heart-dipole problem, $l = 8$

$$\begin{aligned} p(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8) &= x_1x_6^3 - 3x_1x_6x_7^2 + x_3x_7^3 - 3x_3x_7x_6^2 \\ &\quad +x_2x_5^3 - 3x_2x_5x_8^2 + x_4x_8^3 - 3x_4x_8x_5^2 \\ &\quad +0.9563453 \end{aligned}$$

$$\begin{aligned} \mathbf{x_1} &= [-.1,.4],\ \mathbf{x_2} = [.4,1],\ \mathbf{x_3} = [-.7,-.4],\ \mathbf{x_4} = [-.7,.4], \\ \mathbf{x_5} &= [.1,.2], \mathbf{x_6} = [-.1,.2],\ \mathbf{x_7} = [-.3,1.1],\ \mathbf{x_8} = [-1.1,-.3] \end{aligned}$$

References

[1] J. Berchtold and A. Bowyer. Robust arithmetic for multivariate Bernstein-form polynomials. *Computer Aided Design*, 32:681–689, 2000.

[2] J. Berchtold, I. Voiculescu, and A. Bowyer. Multivariate Bernstein form polynomials. Technical Report 31/98, School of Mechanical Engineering, University of Bath, 1998.

[3] G. Farin. *Curves and surfaces in computer aided geometric design.* Academic Press, San Diego, 1993.

[4] R. T. Farouki and V. T. Rajan. On the numerical condition of polynomials in Bernstein form. *Computer Aided Geometric Design*, 4:191–216, 1987.

[5] J. Garloff. The Bernstein algorithm. *Interval Computation*, 2:154–168, 1993.

[6] J. Garloff. The Bernstein expansion and its applications. *Journal of the American Romanian Academy*, (25-27):80–85, 2003. This is a tutorial paper.

[7] 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.*, pages 339–352, London, 1999.

[8] J. Garloff and A. P. Smith. Solution of systems of polynomial equations by using Bernstein expansion. In G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto, editors, *Symbolic Algebraic Methods and Verification Methods*, pages 87–97. Springer, New York, 2001.

[9] S. Malan, M. Milanese, M. Taragna, and J. Garloff. B3 algorithm for robust performance analysis in presence of mixed parametric and dynamic perturbations. In *Proc. of the 31st Conference on Decision and Control*, pages 128–133, Tucson, Arizona, 1992.

[10] R. E. Moore. *Methods and applications of interval analysis.* SIAM, Philadelphia, 1979.

[11] 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*, 04(4):342–352, 2007.

[12] D. Ratz and T. Csendes. On the selection of subdivision directions in interval branch-and-bound methods for global optimization. *Journal of Global Optimization*, 7:183–207, 1995.

[13] M. R. Spencer. *Polynomial real root finding in Bernstein form.* PhD thesis, Department of Civil Engineering, Brigham Young University, Provo, UT, USA, 1994.

[14] V. Stahl. *Interval methods for bounding the range of polynomials and solving systems of nonlinear equations.* PhD thesis, Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria, 1995.

[15] Sun Microsystems, Palo Alto, CA, USA. *Forte FORTRAN 95 User Manual*, 2001.

[16] J. Verschelde. The PHC pack, the database of polynomial systems. Technical report, University of Illinois, Mathematics Department, Chicago, U.S.A., 2001.

[17] M. N. Vrahatis, D. G. Sotiropoulos, and E. C. Triantafyllou. Global optimization for imprecise problems. In I. M. Bomze, T. Csendes, R. Horst, and P. M. Pardalos, editors, *Developments in Global Optimization*, pages 37–54. Kluwer, The Netherlands, 1997.

[18] L. S. Zhang, C. K. Ng, D. Li, and W. W. Tian. A new filled function method for global optimization. *Journal of Global Optimization*, 28:17–43, 2004.