Universität Konstanz

Rigorous Affine Lower Bound Functions for Multivariate Polynomials and Their Use in Global Optimisation

Jürgen Garloff
Andrew P. Smith

Konstanzer Schriften in Mathematik und Informatik

Nr. 250, Juni 2008

ISSN 1430-3558

Rigorous Affine Lower Bound Functions for Multivariate Polynomials and Their Use in Global Optimisation

Jürgen Garloff* and Andrew P. Smith**
*University of Konstanz, Department of
Mathematics and Statistics, Konstanz, Germany
**University of Applied Sciences / HTWG Konstanz,
Department of Computer Science,
Postfach 100543, D-78405 Konstanz, Germany
{garloff,smith}@htwg-konstanz.de

Abstract

This paper addresses the problem of finding tight affine lower bound functions for multivariate polynomials, which may be employed when global optimisation problems involving polynomials are solved with a branch and bound method. These bound functions are constructed by using the expansion of the given polynomial into Bernstein polynomials. The coefficients of this expansion over a given box yield a control point structure whose convex hull contains the graph of the given polynomial over the box. We introduce a new method for computing tight affine lower bound functions based on these control points, using a linear least squares approximation of the entire control point structure. This is demonstrated to have superior performance to previous methods based on a linear interpolation of certain specially chosen control points. The problem of how to obtain a verified affine lower bound function in the presence of uncertainty and rounding errors is also considered. Numerical results with error bounds for a series of randomly-generated polynomials are given.

Key words: Constrained global optimisation, relaxation, affine bound functions, Bernstein polynomials, linear least squares

1 Introduction

Consider the constrained global optimisation problem

$$\min_{x \in F} f(x), \tag{1}$$

where the set of *feasible solutions* F is defined by

$$F := \left\{ x \in S \;\middle|\; \begin{array}{l} g_i(x) \leq 0 \;\text{ for }\; i = 1, \ldots, t \\ x \in X \end{array} \right\},$$

and where $S \subseteq \mathbf{R}^n$, X is an axis-aligned box contained in S, and f, g_i are real-valued functions defined on S. The optimisation problem

$$\min_{x \in R} \underline{f}(x) \tag{2}$$

is termed a *relaxation* of (1) if its set of feasible solutions R fulfils $F \subseteq R$ and $\underline{f}(x) \leq f(x)$ holds for all $x \in F$.

A relaxation may be generated by replacing the functions f, g_i $(i = 1, \ldots, t)$ by lower bound functions $\underline{f}$, $\underline{g}_i$, respectively. Its solution provides a lower bound for the solution of (1). Such relaxations are commonly used in a branch and bound framework for solving problem (1).

Computing a good quality convex lower bound function for a given function is thus of great importance in global optimisation when a branch and bound approach is used. Of special interest are convex envelopes, i.e., uniformly best underestimating convex functions, cf. [7], [19]. Because of their simplicity and ease of computation, constant and affine lower bound functions are especially useful. Constant bound functions are thoroughly used when interval computation techniques are applied to global optimization, e.g. [12], [15]. However, when using constant bound functions, all information about the shape of the given function is lost. A compromise between convex envelopes, which require in the general case a great deal of computational effort, and constant lower bound functions are affine lower bound functions.

To generate an affine relaxation for problem (1), the functions f, g_i $(i = 1, \ldots, t)$ are replaced by affine lower bound functions $\underline{f}$, $\underline{g}_i$, respectively. Then the relaxed problem (2) with the respective set of feasible solutions yields a linear programming problem whose solution provides a lower bound for the solution of (1).

We address the use of relaxations for global optimisation problems (1) in which the objective function f and the functions defining the constraints g_i $(i = 1, \ldots, t)$ are all multivariate polynomials. In this case, affine bound functions can be constructed by utilising the coefficients of the expansion of the given polynomial into Bernstein polynomials.

The organisation of our paper is as follows. Properties of Bernstein polynomials are introduced in Section 2; the reader is also referred to [3], [8], [16], [20]. In Section 3 we present a brief overview of existing methods and a new method for computing Bernstein–based affine bound functions. Section 4 consists of results for a series of randomly-generated polynomials, with a comparison of the error bounds. In Section 5 we present a rigorous version of our new method which will deliver a guaranteed affine lower bound function in the presence of either uncertainty or rounding errors. Directions for future work conclude this paper.

2 Bernstein Expansion

In this section we recall some properties of the Bernstein expansion which are fundamental to our approach.

Firstly, some notation is introduced. We define multiindices $i = (i_1, \ldots, i_n)^T$ as vectors, where the n components are nonnegative integers. The vector 0 denotes the multiindex with all components equal to 0, which should not cause ambiguity. Comparisons are used entrywise. Also the arithmetic operators on multiindices are defined componentwise such that $i \odot l := (i_1 \odot l_1, \ldots, i_n \odot l_n)^T$, for $\odot = +, -, \times$, and $/$ (with $l > 0$). For instance, i/l, $0 \leq i \leq l$, defines the Greville abscissae. For $x \in \mathbf{R}^n$ its multipowers are

$$x^i := \prod_{\mu=1}^{n} x_\mu^{i_\mu}. \tag{3}$$

For the sum we use the notation

$$\sum_{i=0}^{l} := \sum_{i_1=0}^{l_1} \cdots \sum_{i_n=0}^{l_n} . \tag{4}$$

The generalised binomial coefficient is defined by

$$\binom{l}{i} := \prod_{\mu=1}^{n} \binom{l_\mu}{i_\mu}. \tag{5}$$

An n-variate polynomial p,

$$p(x) = \sum_{i=0}^{l} a_i x^i, \quad x = (x_1, \ldots, x_n), \tag{6}$$

can be represented over $I = [0, 1]^n$ as

$$p(x) = \sum_{i=0}^{l} b_i B_i(x), \tag{7}$$

where B_i is the *ith Bernstein polynomial of degree l*

$$B_i(x) = \binom{l}{i} x^i (1 - x)^{l-i} \tag{8}$$

and the so-called *Bernstein coefficients* b_i are given by

$$b_i = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{l}{j}} a_j, \quad 0 \leq i \leq l. \tag{9}$$

We may consider the case of the unit box I without loss of generality, since any nonempty box in $\mathbf{R}^n$ can be mapped affinely thereupon.

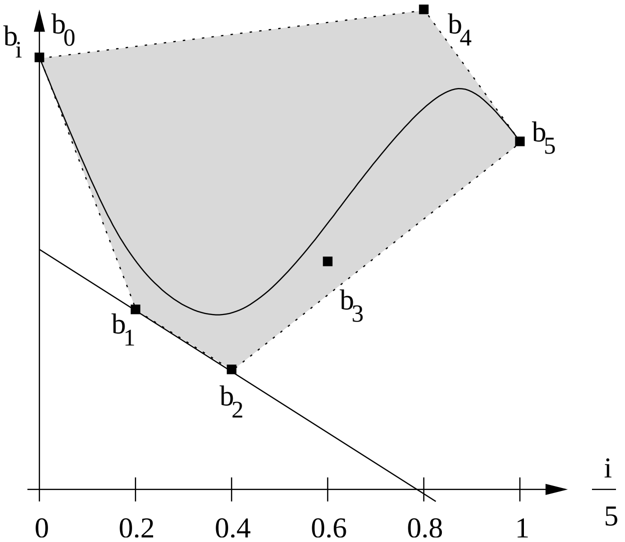

Figure 1: The curve of a polynomial of fifth degree, the convex hull (shaded) of its control points (marked by squares), and an example affine lower bound function.

From (9) the *endpoint interpolation property*

$$b_0 = p(0, \dots, 0) \quad \text{and} \quad b_l = p(1, \dots, 1) \tag{10}$$

follows.

A fundamental property for our approach is the *convex hull property*

$$\left\{ \begin{pmatrix} x \\ p(x) \end{pmatrix} : x \in I \right\} \subseteq conv \left\{ \begin{pmatrix} i/l \\ b_i \end{pmatrix} : 0 \leq i \leq l \right\}, \tag{11}$$

where the convex hull is denoted by *conv*; the points generated by the Bernstein coefficients appearing on the right hand side are called *control points*.

Figure 1 illustrates the convex hull property and the straighforward construction of an affine lower bound function based upon the convex hull of a univariate polynomial of degree 5. The lower bound function passes through the control point associated with the minimum Bernstein coefficient b_2.

3 Affine Lower Bound Functions

There are numerous possible approaches to deriving a tight affine lower bound function from the Bernstein control points (coefficients) of a given polynomial. Methods are introduced in [9], [10], [11] and compared in [11].

The simplest method is to use constant bound functions obtained by choosing the minimum Bernstein coefficient. Other methods rely on a control point associated with the minimum Bernstein coefficient and a determination of n other control points. These are chosen in such a way that the linear interpolant of these points coincides with one of the lower facets of the convex hull of the control points and therefore constitutes a lower bound function for the given polynomial, cf. (11). Such a lower bound function can be obtained as the solution of a linear programming problem [9]. An alternative [10] is the construction of a sequence of hyperplanes passing through a minimum control point which approximate from below the lower part of the convex hull increasingly well. In this case, only the solution of a system of linear equations together with a sequence of back substitutions is needed. The last approach is shown in [11] to have the best performance, in terms of error (the maximum distance of the affine function below the graph of the polynomial) and in computing time. It is compared in Section 4 with the new method presented in the next subsection.

3.1 New Method

An alternative approach to the prior methods is to derive an affine approximation to the *whole set* of control points (and thereby the graph of the polynomial) over the box. In this new method, we propose the use of the linear least squares approximation. This yields an affine function which closely approximates the graph of the polynomial over the whole of the box. It must be adjusted by a downward shift so that it passes under all the control points, yielding a valid lower bound function.

The algorithm includes the following three steps:

1. Compute the Bernstein coefficients b_i, $0 \leq i \leq l$, of p over the box I. There are $m = \prod_{i=1}^{n}(l_i+1)$ such coefficients. We recall, for the convenience of the reader, from [8], cf. [20], an algorithm for the efficient computation of these coefficients, which avoids the calculation of the binomial coefficients and products inherent in (9). It requires $O(n\hat{l}^{n+1})$ additions, where $\hat{l} := \max_{i=1,\ldots,n}\{l_i\}$.

   - $b_i := \frac{a_i}{\binom{l}{i}}$, $i = 0, \ldots, l$.
   - For $r = 1, \ldots, n$:
     - For $k = 1, \ldots, l_r$:

       $$b_i^* := \left\{ \begin{array}{ll} b_i, & \text{if } i_r < k \\ b_i + b_{i_1,\ldots,i_r-1,\ldots,i_n}, & \text{if } i_r \geq k \end{array} \right\}, \; i = 0, \ldots, l.$$

       $$b_i := b_i^*, \; i = 0, \ldots, l.$$

   - b_i are the desired Bernstein coefficients, $i = 0, \ldots, l$.

2. Let A be the $m \times (n+1)$ matrix whose i,jth element is defined as

   $$a_{i,j} = i_j / l_j, \quad \text{for} \quad 1 \leq j \leq n, \quad a_{i,n+1} = 1.$$

Let b be the vector consisting of the corresponding m Bernstein coefficients. Then the coefficients of the linear least squares approximation of all the control points are given by the solution γ to

$$A^T A\gamma = A^T b,$$

yielding the affine function

$$c^*(x) = \sum_{i=1}^{n} \gamma_i x_i + \gamma_{n+1}.$$

For numerically reliable approaches to solving the linear least squares problem, see, e.g., [1].

3. Compute the maximum positive discrepancy between c^* and the control points, and perform a downward shift:

$$\delta^+ = \max\left\{ c^*\left(\frac{i}{l}\right) - b_i \ : \ 0 \leq i \leq l \right\},$$

$$c(x) = c^*(x) - \delta^+, \ x \in I.$$

By construction, c is then a valid affine lower bound function.

Theorem 1 *The following error bound is valid:*

$$0 \leq p(x) - c(x) \leq \max_{i=0,\ldots,l} \left(b_i - c\left(\frac{i}{l}\right) \right), \ x \in I.$$

Proof: It can be seen from (9) that the value of p at a vertex of I coincides with the respective Bernstein coefficient, i.e.,

$$b_i = p\left(\frac{i}{l}\right), \ \text{ for all } \ i = 0, \ldots, l \ \text{ with } \ i_\mu \in \{0, l_\mu\}, \ \mu = 1, \ldots, n.$$

This property divides the surface of the convex hull of the control points into a lower and an upper part, in a natural way. The function describing the upper surface is piecewise affine over the grid given by the Greville abscissae and is denoted by u. The discrepancy $u - c$ is the difference of a piecewise affine and an affine function and must therefore assume its maximum at a point $\frac{i^*}{l}$ at which the associated control point is an exposed vertex of the convex hull, hence $u\left(\frac{i^*}{l}\right) = b_{i^*}$. So we may conclude

$$\begin{array}{rcl} \max_{x \in I} \left(p(x) - c(x)\right) & \leq & \max_{x \in I} \left(u(x) - c(x)\right) \\ & = & \max_{i=0,\ldots,l} \left(u\left(\frac{i}{l}\right) - c\left(\frac{i}{l}\right)\right) \\ & = & \max_{i=0,\ldots,l} \left(b_i - c\left(\frac{i}{l}\right)\right). \qquad \square \end{array}$$

The results presented below in Section 4 show that, in general, the lower bound functions delivered by the new method fit the given polynomial more

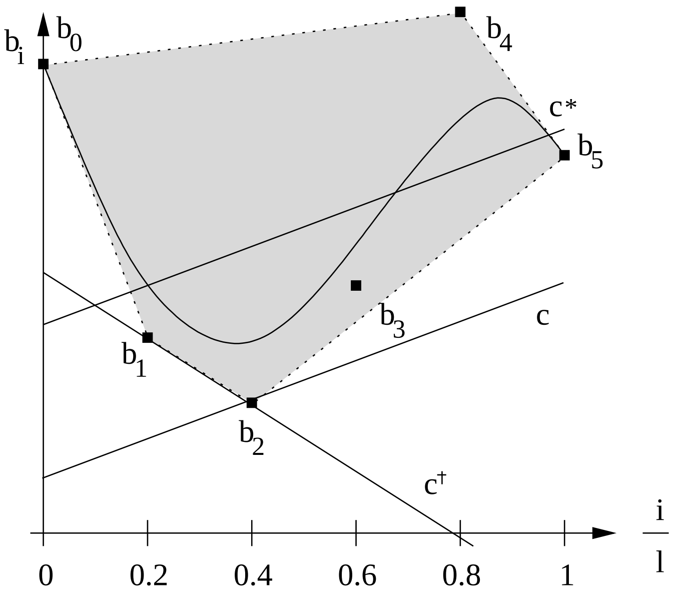

Figure 2: The curve of the polynomial from Fig. 1 (bold), the convex hull (shaded) of its control points (marked by squares), and the affine lower bound functions $c^\dagger$ and c.

closely than those given by the previous methods. We may conclude that the approach of approximating all of the control points is superior to the technique of interpolating $n+1$ control points which form a lower facet of the convex hull. In other words, the general shape information of the polynomial over the whole of the box would appear to be more valuable than just the local behaviour of the polynomial near its minimum over the box.

This is illustrated by Figure 2, which shows the same univariate polynomial of degree 5 as in Figure 1. In this case the lower bound function c from the new method more closely represents the polynomial behaviour than $c^\dagger$ from the previous methods.

4 Results with Randomly Generated Polynomials

The new method was tested with a number of multivariate polynomials (6) in n variables with degree $l = (D, \ldots, D)^T$ and compared to constant bound functions and a previous method [10]. The polynomials were given k non-zero terms, with the non-zero coefficients being randomly generated with $a_i \in [-1, 1]$.

Table 1 lists the results for different values of n, D, and k; $(D+1)^n$ is the number of Bernstein coefficients. In each case 100 random polynomials were generated and the mean computation time and error δ are given. An upper

Table 1: Results for randomly generated polynomials.

Method				Constant bound functions		Previous method		New method	
n	D	k	$(D+1)^n$	time (s)	δ	time (s)	δ	time (s)	δ
2	2	5	9	0.000002	1.420	0.000069	0.981	0.000006	0.698
2	6	10	49	0.000011	2.002	0.00031	1.677	0.000024	1.496
2	10	20	121	0.000044	2.852	0.00074	2.511	0.000070	2.435
4	2	20	81	0.000053	3.458	0.0012	2.797	0.000090	2.468
4	4	50	625	0.00055	5.682	0.0093	5.045	0.00079	4.870
6	2	20	729	0.00056	4.075	0.016	3.353	0.0010	3.131
8	2	50	6561	0.0090	6.941	0.24	6.291	0.018	6.300
10	2	50	59049	0.11	7.142	3.43	6.503	0.29	6.473
12	2	50	531441	1.32	7.377	$>$ 1 minute		3.81	6.712

bound on the discrepancy between the polynomial and its lower bound function over I is computed as

$$\delta = \max_i \left\{ b_i - d\left(\frac{i}{l}\right) \right\},$$

where d is $c^\dagger$ or c, respectively. The results were produced with `C++` on a 2.4 GHz PC.

Compared to the previous method mentioned at the beginning of Section 3, the new method described in Subsection 3.1 in general delivers tighter bound functions, and is one to two orders of magnitude faster. Compared to constant bound functions, it provides much better bound functions, but is only slower by a factor of 1.4 to 3. Given this relatively small factor, it is expected that further improvements which rely on *all* Bernstein coefficients will likely be marginal; cf., however, Section 6.

5 Rigorous Bound Functions

Due to rounding errors, inaccuracies may be introduced into the calculation of the Bernstein coefficients and the corresponding bound functions. As a result, the computed affine function may not stay below the given polynomial over the box of interest. We also wish to consider the case of uncertain (interval) input data. In either case, it is often desirable to compute the affine lower bound functions in such a way that it can be *guaranteed* to stay below the given polynomial. Methods based on Bernstein expansion can be adjusted relatively easily to work with interval data, and the safe interpolation of interval control points can be facilitated by a method similar to that introduced in [2]; see also [13].

The method presented in this paper is especially well suited to this purpose and is easily adapted into a verified version using interval arithmetic. For an introduction into interval arithmetic the reader is referred to, e.g., [15].

1. Given a polynomial with interval coefficients (which may result either from some uncertainties in the problem, or as very small intervals of machine precision width, in order to cater for rounding errors), compute interval Bernstein coefficients, using the same procedure as before in step 1, but with interval arithmetic.
2. Compute the linear least squares approximation of the control points as before, except using the *midpoints* of the interval Bernstein coefficients.
3. Compute the discrepancy δ^+ and perform the downward shift as before, but according to the *lower bounds* of the control points (Bernstein coefficients).

Step 2 (the bulk of the computation) does not need to be performed rigorously, and is implemented with normal floating point arithmetic. Since only the first and last steps need to be performed with interval arithmetic, little extra computational effort is required.

6 An Alternative Basis

In the univariate case, a basis which is related to the basis of the Bernstein polynomials was presented in [5]. This basis (henceforth termed the DP basis) is defined as follows

$$\begin{aligned}
&C_0(x) = (1-x)^l,\\
&C_i(x) = x(1-x)^{l-i}, \quad 1 \le i \le \left\lfloor \tfrac{l}{2} \right\rfloor - 1,\\
&C_i(x) = x^i(1-x), \quad \left\lfloor \tfrac{l+1}{2} \right\rfloor + 1 \le i \le l-1,\\
&C_l(x) = x^l.
\end{aligned}$$

In addition, if l is even,

$$C_{\frac{l}{2}}(x) = 1 - x^{\frac{l}{2}+1} - (1-x)^{\frac{l}{2}+1}$$

and, if l is odd,

$$\begin{aligned}
&C_{\frac{l-1}{2}}(x) = x(1-x)^{\frac{l+1}{2}} + \tfrac{1}{2}\left[1 - x^{\frac{l+1}{2}} - (1-x)^{\frac{l+1}{2}}\right],\\
&C_{\frac{l+1}{2}}(x) = \tfrac{1}{2}\left[1 - x^{\frac{l+1}{2}} - (1-x)^{\frac{l+1}{2}}\right] + x^{\frac{l+1}{2}}(1-x).
\end{aligned}$$

The DP basis shares with the Bernstein basis many properties which are important in Computer Aided Geometric Design, e.g., the endpoint interpolation property (10), the convex hull property (11), and the variation diminishing property. The latter means that the maximum number of intersections of a straight line with the polynomial curve is not greater than the number of intersections with the polygon which connects the control points. The transformation between the two bases was given in [14], cf. also [4]. From this transformation it can be seen that for some l, e.g. $l = 3$ or $l = 5$, the convex hull of the control points w.r.t. the Bernstein basis is contained within the convex hull of the control points w.r.t. the DP basis. So this basis is only advantageous for the construction of affine lower bound functions if an algorithm could be designed for the computation of the coefficients of a given polynomial w.r.t. the DP basis which exhibits linear complexity (in contrast to our algorithm, given in the beginning of Subsection 3.1, which exhibits quadratic complexity in the univariate case). The design of such an algorithm is an open problem.

7 Future Work

- The bound functions are to be integrated as a black-box component of a general-purpose package for the solution of global optimisation and continuous constraint satisfaction problems, part of the EU-funded COCONUT project [17].

- The present limitation of our approach is the exponential growth in the number of Bernstein coefficients which are computed in Step 1 of our algorithm. Compared with this, the computational effort of Steps 2 and 3 is negligible. However it is worth noting that the technique will already handle many types of polynomial constraint and objective functions frequently arising in optimization problems. Typically, such polynomials are sparse, with not all variables appearing, and the degree in most variables is low.

  For constant bound functions, significant progress towards the reduction of the computational complexity has recently been achieved in [18]. By means of results concerning the Bernstein coefficients of monomials and through the development of an implicit Bernstein form, an efficient means for the computation of the range of a polynomial was given. Many examples from literature show that often the determination of only a small number of Bernstein coefficients is required, thereby reducing the computational complexity by orders of magnitude. It is intended to extend these results to the efficient calculation of affine bound functions shortly.

- Our approach may be extended to the construction of affine lower bound functions for arbitrary sufficiently differentiable functions, by using Taylor expansion. A high-degree Taylor polynomial can be computed (the introduction of higher degree polynomial terms is not necessarily problematic for our approach), for which Bernstein coefficients and a bound function can be computed, as before. The remainder of the Taylor expansion can be enclosed in an interval, by using established methods from interval analysis, e.g. [6]. Subtracting this interval from the lower bound function of the Taylor polynomial provides the lower bound function for the given function.

Acknowledgement

This work has been supported by the German Research Council (DFG) and the State of Baden-Württemberg, Germany.

References

[1] Björck A. (2004) "The calculation of linear least squares problems," in *Acta Numerica 2004*, A. Iserles et al., Eds., Cambridge University Press, 1–53.

[2] Borradaile G. and Van Hentenryck P. (2005), "Safe and tight linear estimators for global optimization," *Mathematical Programming Vol. 102 (3)*, 495–518.

[3] Cargo G. T. and Shisha O. (1966), "The Bernstein form of a polynomial," *J. Res. Nat. Bur. Standards Vol. 70B*, 79–81.

[4] Dejdumrong N. (2006), "Rational DP-ball curves," in "International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)," 478–483.

[5] Delgado J. and Peña J. M. (2003), "A shape preserving representation with an evaluation algorithm of linear complexity," *Computer Aided Geometric Design Vol. 20*, 1–20.

[6] Eble I. and Neher M. (2003), "ACETAF: A software package for computing validated bounds for Taylor coefficients of analytic functions," *ACM Trans. on Math. Software Vol. 29*, 263–286.

[7] Floudas C. A. (2000), "Deterministic Global Optimization: Theory, Methods, and Applications," *Series Nonconvex Optimization and its Applications Vol. 37*, Kluwer Acad. Publ., Dordrecht, Boston, London.

[8] Garloff J. (1986), "Convergent bounds for the range of multivariate polynomials," *Interval Mathematics 1985,* K. Nickel, editor, *Lecture Notes in Computer Science Vol. 212*, Springer, Berlin, 37–56.

[9] Garloff J., Jansson C. and Smith A. P. (2003), "Lower bound functions for polynomials," *J. Computational and Applied Mathematics Vol. 157*, 207–225.

[10] Garloff J. and Smith A. P. (2004), "An improved method for the computation of affine lower bound functions for polynomials," in "Frontiers in Global Optimization," Floudas C. A. and Pardalos P. M., Eds., *Series Nonconvex Optimization with its Applications Vol. 74*, Kluwer Acad. Publ., Dordrecht, Boston, London, 135–144.

[11] Garloff J. and Smith A. P. (2005), "A comparison of methods for the computation of affine lower bound functions for polynomials," in "Global Optimization and Constraint Satisfaction," Jermann C., Neumaier A. and Sam D., Eds., *Lecture Notes in Computer Science*, No. 3478, Springer-Verlag, Berlin, Heidelberg, 71–85.

[12] Hansen E. R. (2003), “Global Optimization Using Interval Analysis,” 2nd edition, Marcel Dekker, Inc., New York.

[13] Hongthong S. and Kearfott R. B. (2004), “Rigorous linear overestimators and underestimators,” submitted to *Mathematical Programming B.*

[14] Jiang S. and Wang G. (2005), “Conversion and evaluation for two types of parametric surfaces constructed by NTP bases,” *Mathematical and Computer Modelling Vol. 49*, 321–329.

[15] Kearfott R. B. (1996), “Rigorous Global Search: Continuous Problems,” *Series Nonconvex Optimization and its Applications Vol. 13*, Kluwer Acad. Publ., Dordrecht, Boston, London.

[16] Prautzsch H., Boehm W. and Paluszny M. (2002), “Bézier and B-Spline Techniques,” Springer, Berlin, Heidelberg.

[17] Schichl H. (2003), “Mathematical Modeling and Global Optimization,” Habilitation thesis, University of Vienna, to appear in Cambridge University Press, available under `http://www.mat.univie.ac.at/~herman/papers.html`.

[18] Smith A. P., “Fast construction of constant bound functions for sparse polynomials,” to appear in *J. Global Optimization.*

[19] Tawarmalani M. and Sahinidis N. V. (2002), “Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications,” *Series Nonconvex Optimization and its Applications Vol. 65*, Kluwer Acad. Publ., Dordrecht, Boston, London.

[20] Zettler M. and Garloff J. (1998), “Robustness analysis of polynomials with polynomial parameter dependency using Bernstein expansion,” *IEEE Trans. Automat. Contr. Vol. 43*, 425–431.