Guaranteed Affine Lower Bound Functions for Multivariate Polynomials

Jürgen Garloff* and **Andrew P. Smith**

University of Applied Sciences (HTWG), Institute of Applied Research, Postfach 100543, D-78405 Konstanz, Germany

1 Introduction

Computing a good quality convex lower bound function for a given function is of great importance in global optimisation when a branch and bound approach is used. 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 optimisation, cf. [6]. However, when using constant bound functions, all information about the shape of the given function is lost. A compromise between convex envelopes, e.g. [1], which require in the general case a great deal of computational effort, and constant lower bound functions are affine lower bound functions. In this paper we show how tight affine lower bound functions for multivariate polynomials can be constructed by utilising the coefficients of the expansion of the given polynomial into Bernstein polynomials. For further details, see [5].

2 Bernstein Expansion

Comparisons and the arithmetic operators on multiindices $i = (i_1, \ldots, i_n)^T$ are defined componentwise. For $x \in \mathbf{R}^n$ its multipowers are $x^i := x_1^{i_1} \ldots x_n^{i_n}$. Using the compact notation $\sum\limits_{i=0}^{l} := \sum\limits_{i_1=0}^{l_1} \ldots \sum\limits_{i_n=0}^{l_n}$, $\binom{l}{i} := \prod\limits_{\mu=1}^{n} \binom{l_\mu}{i_\mu}$, an n-variate polynomial p, $p(x) = \sum\limits_{i=0}^{l} a_i x^i$, $x \in I = [0,1]^n$, can be represented as $p(x) = \sum\limits_{i=0}^{l} b_i B_i(x)$, where $B_i(x) = \binom{l}{i} x^i (1-x)^{l-i}$ is the ith Bernstein polynomial of degree l, and the so-called *Bernstein coefficients* b_i are given by $b_i = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{l}{j}} a_j$, $0 \le i \le l$. For an efficient computation of the Bernstein coefficients, see [9].

A fundamental property for our approach is the *convex hull property* ($conv$ denoting the convex hull)

$$\left\{ \begin{pmatrix} x \\ p(x) \end{pmatrix} : x \in I \right\} \subseteq conv \left\{ \begin{pmatrix} i/l \\ b_i \end{pmatrix} : 0 \le i \le l \right\}. \tag{1}$$

3 Affine Lower Bound Functions

There are numerous possible approaches to deriving a tight affine lower bound function from the Bernstein control points of a given polynomial, the convex hull of which is formed in (1). Methods are introduced in [2], [3], [4] and compared in [4]. 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 (cf. (1)) and therefore constitutes a lower bound function for the given polynomial. Such a bound function can be obtained by the solution of a system of linear equations together with a sequence of back substitutions [3].

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 I. Here 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:

1. Let A be the $m \times (n+1)$ matrix whose i,jth element is defined as $a_{i,j} = i_j / l_j$, for $1 \le j \le n$, $a_{i,n+1} = 1$. Let b be the vector consisting of the corresponding m Bernstein coefficients. Then the coefficients γ of the affine function can be obtained by fitting it over the control points using the least squares method: $A^T A \gamma = A^T b$, yielding the affine function $c^*(x) = \sum_{i=1}^{n} \gamma_i x_i + \gamma_{n+1}$.
2. Compute the maximum positive discrepancy between c^* and the control points, and perform a downward shift: $\delta^+ = \max \left\{ c^* \left(\frac{i}{l} \right) - b_i : 0 \le i \le l \right\}$. By construction, c, $c(x) = c^*(x) - \delta^+,\ x \in I$, is a valid affine lower bound function.

* Corresponding author: e-mail: garloff@htwg-konstanz.de, Phone: +49 7531 206 627, Fax: +49 7531 206 559

Table 1 Results for randomly generated polynomials.

Method				Constant bound functions		Facet method		Least squares 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

Theorem 3.1 *The following error bound is valid:* $0 \le p(x) - c(x) \le \max_{i=0,\ldots,l} \left(b_i - c\left(\frac{i}{l}\right)\right)$, $x \in I$.

4 Results with Randomly Generated Polynomials

The new method was tested with numerous polynomials in n variables with degree $l = (D, \ldots, D)^T$ and compared to constant bound functions and a previous method [3], termed *facet method*. Coefficients were randomly generated in $[-1, 1]$.

Table 1 lists the results for different values of n, D, and k (the number of non-zero terms); $(D+1)^n$ is the number of Bernstein coefficients. In each case 100 random polynomials were generated and the mean computation time and error δ are given, where δ is an upper bound on the discrepancy between the polynomial and its lower bound function over I, computed as the right-hand side of the inequality in Theorem 3.1. The results were produced with C++ on a 2.4 GHz PC; for details on the software used, see [7].

Compared to the previous method, the new method 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. For a method for computing constant bound functions for sparse polynomials which avoids the exponential complexity of the approach presented in this paper, the reader is referred to [8].

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. We also wish to consider the case of uncertain (interval) input data. The method presented in this paper is especially well suited to the computation of the affine lower bound function in such a way that it can be *guaranteed* to stay below the given polynomial and is easily adapted into a verified version using interval arithmetic (e.g. [6]); for a technical report on this software, see [7].

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), firstly compute the interval Bernstein coefficients. Then compute the linear least squares approximation of the control points as before, except using the *midpoints* of the interval Bernstein coefficients. Note that this step (the bulk of the computation) does not need to be performed rigorously, and is implemented with normal floating point arithmetic. Lastly, compute the discrepancy δ^+ and perform the downward shift as before, but according to the *lower bounds* of the Bernstein coefficients.

Acknowledgements This work has been supported by the German Research Council (DFG).

References

[1] C. A. Floudas, Deterministic Global Optimization: Theory, Methods, and Applications, (Kluwer Acad. Publ., Dordrecht, 2000).
[2] J. Garloff, C. Jansson, and A. P. Smith, Lower bound functions for polynomials, J. Comp. and Appl. Math. Vol. 157 (2003), pp. 207–225.
[3] J. Garloff and A. P. Smith, An improved method for the computation of affine lower bound functions for polynomials, in: Frontiers in Global Optimization, edited by C. A. Floudas and P. M. Pardalos, (Kluwer Acad. Publ., Dordrecht, 2004) pp. 135–144.
[4] J. Garloff and A. P. Smith, A comparison of methods for the computation of affine lower bound functions for polynomials, in: Global Optimization and Constraint Satisfaction, edited by C. Jermann, A. Neumaier, and D. Sam, Lecture Notes in Computer Science No. 3478, (Springer-Verlag, Berlin, Heidelberg, 2005) pp. 71–85.
[5] J. Garloff and A. P. Smith, Rigorous affine lower bound functions for multivariate polynomials and their use in global optimisation, Internal report, HTWG Konstanz (2007).
[6] R. B. Kearfott, Rigorous Global Search: Continuous Problems, (Kluwer Acad. Publ., Dordrecht, 1996).
[7] A. P. Smith, BeBP: A software package for the rigorous computation of affine bound functions for polynomials, submitted to ACM Trans. on Math. Software (2006).
[8] A. P. Smith, Fast construction of constant bound functions for sparse polynomials, to appear in J. Global Optimization (2007).
[9] M. Zettler and J. Garloff, Robustness analysis of polynomials with polynomial parameter dependency using Bernstein expansion, IEEE Trans. Automat. Contr. Vol. 43 (1998), pp. 425–431.