Polytope-Based Computation of Polynomial Ranges

Christoph Fünfzig*
LE2I, UMR CNRS 5158,
Univ. Bourgogne,
21078 Dijon, France
Christoph.Fuenfzig@u-bourgogne.fr

Dominique Michelucci
LE2I, UMR CNRS 5158,
Univ. Bourgogne,
21078 Dijon, France
dmichel@u-bourgogne.fr

Sebti Foufou
LE2I, UMR CNRS 5158,
Univ. Bourgogne, Dijon, Fr
CSE, Qatar University,
P.O.B. 2713, Doha, Qatar
sfoufou@u-bourgogne.fr,
sfoufou@qu.edu.qa

ABSTRACT

Polynomial ranges are commonly used for numerically solving polynomial systems with interval Newton solvers. Often ranges are computed using the convex hull property of the tensorial Bernstein basis, which is exponential size in the number n of variables.

In this paper, we consider methods to compute tight bounds for polynomials in n variables by solving two linear programming problems over a polytope. We formulate several polytopes based on the tensorial Bernstein basis, and we formulate a polytope for the quadratic patch $\mathcal{Q}_n := (x_1, \dots, x_n, x_1^2, \dots, x_n^2, x_1x_2, \dots, x_{n-1}x_n)$ by projections. This Bernstein polytope has $\Theta(n^2)$ hyperplanes. We give the number of vertices, the number of hyperplanes, and the volume of each polytope for $n = 1,2,3,4$, and we compare the range widths computed with all of them for random n-variate polynomials with $n \leq 10$. The Bernstein polytope of polynomial size gives only marginally worse range bounds compared to the range bounds obtained with the tensorial Bernstein basis of exponential size.

Categories and Subject Descriptors

G.1.5 [**Numerical Analysis**]: Roots of Nonlinear Equations; G.1.6 [**Numerical Analysis**]: Optimization

Keywords

polynomial ranges, Bernstein polynomials, multivariate polynomials, polytopes

1. INTRODUCTION

Ranges of polynomials and properties of tensorial Bernstein bases are typically used for numerically solving polynomial systems with interval Newton solvers. These solvers require tight ranges of the Newton map $N(\mathbf{x}) = \mathbf{x} - P(\mathbf{x})(P'(\mathbf{m}))^{-1}$, containing multivariate polynomials over an n-dimensional domain $\mathcal{D}$ with center $\mathbf{m}$. Popular methods in CAD-CAM for intersecting Bézier or piecewise algebraic curves and surfaces are examples for them.

In this paper, we consider the problem of computing a tight range, *i.e.,* a lower bound and an upper bound, of a given multivariate polynomial $p(\mathbf{x})$, $\mathbf{x} \in [0,1]^n$ with total degree of at most 2.

Several authors proposed to use properties of the tensorial Bernstein basis for computing ranges of polynomials $p(\mathbf{x})$ with $\mathbf{x} \in [0,1]^n$. The general case $\bar{\mathbf{x}}$ in $[\mathbf{a},\mathbf{b}] = \{\mathbf{x} \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i\}$ can be reduced to the case with $\mathbf{x}$ in $[0,1]^n$ by scaling the variables $\bar{x}_i = a_i + x_i(b_i - a_i)$. The polynomial is expressed in the tensorial Bernstein basis, and due to the convex hull property, the smallest coefficient is a lower bound, and the largest is an upper bound for the range. This method can be improved, when it is considered as polytope-based. With *polytope-based* methods, we denote the class of methods, which solve for the lower bound and for the upper bound with linear programming over a polytope. The polytope is a bounded, convex set containing the feasible points for the linear programming problem. Different polytopes of different dimension can be considered, and the aim of this article is to define and compare them. As far as we know, it is the first article, which discusses polytopes related to Bernstein polynomials. Polytopes related to Chebyshev polynomials have been used in the PhD thesis of O. Beaumont [1].

For conciseness, non-polytope-based methods are not discussed, and the reader is referred to [8] for a presentation and a comparison of interval methods for computing ranges of bivariate polynomials.

We consider only quadratic polynomials here. In principle, all arbitrary degree polynomials can be reduced to quadratic systems using auxilliary variables and equations, although the performance implications of this are yet to be analyzed. Large classes of geometric constraint systems are quadratic. Furthermore, it is possible to extend the polytopes defined in this article to higher degrees.

The paper is organized as follows. Section 2 gives the definitions of tensorial Bernstein bases and introduces the reader to the complexity issues of range computation. Section 3 presents the standard convex hull method for computing polynomial ranges and provides a polytope-based formulation. It is possible to shrink this polytope, and Section 4 presents the convex hull method based on this shrinked polytope. Section 5 uses the convex hull of the control points of the quadratic patch, defined by the canonical basis functions. All these polytopes have an exponential complexity in terms of n: both their number of vertices and their number of hyperplanes is at least exponential in the number n of variables. Section 6 presents a polytope with a polynomial number of hyperplanes, derived from the polytope in Section 5 by projections. Using linear programming in polynomial time, this polytope permits to com-

*The first author would like to thank the Conseil Régional de Bourgogne for support in a postdoc fellowship.

pute tight ranges of polynomials, which are sharper than the ones provided by interval analysis.

2. COMPLEXITY ISSUES

According to a theorem by Gaganov [5], computing an ε-accurate range of a polynomial is NP-hard. A consequence is that there is no geometric basis for quadratic polynomials providing tight ranges: the polynomials of such a geometric basis are non-negative for $\mathbf{x} \in [0,1]^n$, and their sum equals 1 for any x. It is possible to derive a geometric basis from a simplex enclosing the patch

$$\mathcal{Q}_n = (x_1,\ldots,x_n,x_1^2,\ldots,x_n^2,x_1x_2,\ldots,x_{n-1}x_n), \quad \mathbf{x} \in [0,1]^n.$$

Assume the smallest simplex enclosing $\mathcal{Q}_n$ is known. Then the smallest and greatest coefficient of a quadratic polynomial in the basis derived from this smallest simplex gives a range of the polynomial. Moreover, this method is polynomial time since there are only $O(n^2)$ coefficients to compute. The obtained range can not be tight, because it would contradict Gaganov's theorem, which establishes the NP-hardness of computing an ε-accurate range of a polynomial.

Linear programming has polynomial time-complexity [9]. In practice, interior-point methods and simplex methods are polynomial time. In some exceptional situations similar to Klee-Minty, the number of iterations can be exponential (2^n), for both kinds of methods [2]. The ellipsoid method is polynomial time in the worst case [9] but it is not competitive in practice. No *strongly* polynomial algorithm is currently known, and it is still unknown whether linear programming is strongly polynomial or not.

2.1 Tensorial Bernstein Bases

The $d+1$ Bernstein polynomials $B_i^{(d)}$ of degree d are a basis for degree-d, univariate polynomials

$$B_i^{(d)}(x) = \binom{d}{i} x^i (1-x)^{d-i}$$

The conversion with the canonical basis $(x^0,x^1,\ldots,x^d)$ is a linear mapping. Classical formulas [3] are

$$x^k = \binom{d}{k}^{-1} \sum_{i=k}^{d} \binom{i}{k} B_i^{(d)}(x)$$

$$x = \frac{1}{d} \sum_{i=1}^{d} i B_i^{(d)}(x)$$

$$x^0 = 1 = \sum_{i=0}^{d} B_i^{(d)}(x) \text{ i.e., their sum equals 1.}$$

DEFINITION 1. *For each univariate polynomial $p(x)$ of degree d, there exists* **an univariate Bernstein representation**, *i.e., coefficients $p_i \in \mathbb{R}$ so that $p(x) = \sum_i p_i B_i^{(d)}(x)$ is a linear combination of the Bernstein basis functions.*

The Bernstein basis functions sum to 1, and every $B_i^{(d)}(x)$ is non-negative for $x \in [0,1]$. These properties imply that for $x \in [0,1]$, $p(x) = \sum_i p_i B_i^{(d)}(x)$ is even a convex combination of the coefficients p_i. For a polynomial $p(x)$ over $x \in [0,1]$, its value $p(x)$ lies in the convex hull of the control points $p_i \in \mathbb{R}$, which is just the interval $[\min p_i, \max p_i]$. This enclosure is tight.

DEFINITION 2. *For control points $\mathbf{p}_i \in \mathbb{R}^n$, $\mathbf{p}(x)$ describes a* **Bézier curve in** $\mathbb{R}^n$, *and the curve $\mathbf{p}(x)$, $x \in [0,1]$ lies inside the convex hull of its control points $\mathbf{p}_i$.*

Since $x = 0 \cdot B_0^{(d)}(x) + 1/d \cdot B_1^{(d)}(x) + 2/d \cdot B_2^{(d)}(x) + \ldots d/d \cdot B_d^{(d)}(x)$, the function graph $(x, y = p(x) = \sum_i p_i B_i^{(d)}(x))$ for $x \in [0,1]$, lies in the convex hull of its control points $(i/d, p_i) \in \mathbb{R}^2$.

For $i = 1,\ldots,d-1$, the maximum of $B_i^{(d)}(x)$ occurs at $x = i/d$ and equals $B_i^{(d)}(i/d)$. Later, this property makes possible some improvements of the enclosures given by the convex hull of control points. In the following, we denote the Bernstein polynomials of degree 2 by $B_0^{(2)}(x) \in [0,1]$, $B_1^{(2)}(x) \in [0,1/2]$, $B_2^{(2)}(x) \in [0,1]$ for $x \in [0,1]$. If the degree superscript is omitted, we assume degree 2 implicitly.

For multivariate polynomials, a Bernstein basis can be constructed using the tensorial product (*TBB*) of univariate Bernstein basis functions

$$(B_0^{(d)}(x_1),\ldots,B_d^{(d)}(x_1)) \times (B_0^{(d)}(x_2),\ldots,B_d^{(d)}(x_2)) \times \ldots$$

For notation, it is convenient to use a multi-index $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n) \in [0:d]^n$.

DEFINITION 3. *For a multi-index $\boldsymbol{\alpha} \in [0:d]^n$, the corresponding* **multivariate Bernstein basis function (of degree d)** *is defined by $B_{\boldsymbol{\alpha}}(\mathbf{x}) := B_{\alpha_1}(x_1)\cdots B_{\alpha_n}(x_n)$. For each multivariate polynomial p of maximum degree d, there exists* **a multivariate Bernstein representation**, *i.e., coefficients $p_{\boldsymbol{\alpha}} \in \mathbb{R}$, $\boldsymbol{\alpha} \in [0:d]^n$ so that $p(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in[0:d]^n} p_{\boldsymbol{\alpha}} B_{\boldsymbol{\alpha}}^{(d)}(\mathbf{x})$ is a linear combination of the Bernstein basis functions.*

The convex hull property extends to the TBB, which provides tight enclosures of multivariate polynomials $p(\mathbf{x})$, $\mathbf{x} \in [0,1]^n$.

In algebraic geometry, the multivariate polynomials of total degree d are very important.

DEFINITION 4. *Let $p(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in[0:d]^n} p_{\boldsymbol{\alpha}} x_1^{\alpha_1}\cdots x_n^{\alpha_n}$ be a multivariate polynomial in $\mathbf{x} = (x_1\ldots,x_n)$. If there is an $\boldsymbol{\alpha}$ with $\alpha_i = d$ and $p_{\boldsymbol{\alpha}} \neq 0$, we say* **the multivariate polynomial** p **has maximum degree** d. *In case $p_{\boldsymbol{\alpha}} = 0$ for all $\boldsymbol{\alpha}$, $\alpha_1+\ldots+\alpha_n > d$, we say* **the multivariate polynomial** p **has total degree** d.

Note that the vector space of quadratic polynomials in n variables $x_1,\ldots,x_n$ (*i.e.,* total degree 2) has dimension $1+n+(n+1)n/2 \in O(n^2)$. This is actually the class of polynomials, we work with in the rest of the paper. But the smallest polynomial space with a TBB, containing them, is the space of polynomials of maximum degree d. Unfortunately, this space (and the tensorial Bernstein basis) has dimension $\Theta((d+1)^n)$, which is exponential in the number n of unknowns, even for linear systems ($d=1$). It is not a problem in CAD-CAM, where n is small (*e.g.,* computing the intersection of three surfaces in 3-space), but the exponential size is a problem for solving polynomial systems with a large number n of variables (*e.g.,* computing the coordinates of points with specified distances).

3. CONVEX HULL RANGE

The range of $p(\mathbf{x})$, $\mathbf{x} \in [0,1]^n$ is given by the convex hull of the control points of p in the TBB

$$(B_0(x_1),B_1(x_1),B_2(x_1)) \times \ldots \times (B_0(x_n),B_1(x_n),B_2(x_n))$$

This simple method is used for example in [6, 8, 10].

The method expresses the polynomial p in the TBB as $p(\mathbf{x}) = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} B_{\boldsymbol{\alpha}}(\mathbf{x})$, where $\boldsymbol{\alpha}$ is a multi-index. The smallest coefficient $p_{\boldsymbol{\alpha}}$ is a lower bound for $p(\mathbf{x})$, $\mathbf{x} \in [0,1]^n$. The greatest coefficient $p_{\boldsymbol{\alpha}}$ is an upper bound for $p(\mathbf{x})$, $\mathbf{x} \in [0,1]^n$.

This method does not need linear programming but it is worth to formulate it as a polytope-based method.

For $n = 1$, one variable $\mathbf{x} = (x_1)$: The algebraic patch $(\lambda_0 = B_0(x_1), \lambda_1 = B_1(x_1), \lambda_2 = B_2(x_1))$, where $x_1 \in [0,1]$, is enclosed in the triangle defined by $\lambda_0 + \lambda_1 + \lambda_2 = 1$ and all λ_i are non-negative. See Figure 1, left.

Similarly, consider $n = 2$ and $\mathbf{x} = (x_1, x_2)$. The algebraic patch

$$\mathcal{B}_2 := (\lambda_{ij} := B_i(x_1)B_j(x_2)),\ (i,j) \in \{0,1,2\}^2,\ \mathbf{x} \in [0,1]^2$$

is enclosed in the simplex with points $(\lambda_{00}, \ldots, \lambda_{22})$, which fulfill $0 \le \lambda_{ij}$ for $(i,j) \in \{0,1,2\}^2$ and $1 = \sum_{i,j} \lambda_{ij}$.

Clearly, $\max \sum_{i,j} p_{ij}\lambda_{ij} = \max p_{ij}$, and $\min \sum_{i,j} p_{ij}\lambda_{ij} = \min p_{ij}$, where p_{ij} is the coordinate of basis function $B_{ij}(x_1, x_2)$.

For $n = 3$ and $\mathbf{x} = (x_1, x_2, x_3)$, the coordinates of the simplex are λ_{ijk} where $(i,j,k) \in \{0,1,2\}^3$. Again, all λ_{ijk} are non-negative and their sum equals 1.

For arbitrary n, the simplex has 3^n vertices, and 3^n hyperfaces, and it is $(3^n - 1)$-dimensional. This method for computing the range of $p(\mathbf{x})$, $\mathbf{x} \in [0,1]^n$ is exponential time. It is not practicable for large n due to the exponential number 3^n of coefficients in the tensorial Bernstein basis. Note that the vector space of quadratic polynomials in n variables $x_1, \ldots, x_n$ has dimension $1 + n + (n+1)n/2 \in O(n^2)$, which is much smaller than 3^n.

This polytope formulation seems to be useless at first glance but it makes possible a new improvement in the next section.

Notice that $\mathcal{B}_2$ contains also polynomials $x_1^2 x_2, x_1^2 x_2^2, x_1 x_2^2$ that are not quadratic anymore. It is possible to formulate these properties explicitly in terms of the p_{ij}. Consider the quadratic polynomial $p(x_1, x_2)$. It is written

$$p(x_1, x_2) = \sum_{i=0}^{2} \sum_{j=0}^{2} c_{ij} x_1^i x_2^j$$

in the canonical basis, with $c_{12} = c_{21} = c_{22} = 0$. In matrix notation $p(x_1, x_2) = (1, x_1, x_1^2) C (1, x_2, x_2^2)^t$ with the 3×3 coefficient matrix $C = (c_{ij})_{i,j=1\ldots3}$ with respect to the canonical basis. In the TBB, $p(x,y) = \sum_{i=0}^{2} \sum_{j=0}^{2} p_{ij} B_i(x_1) B_j(x_2)$ has matrix notation

$$p(x_1, x_2) = (B_0(x_1),\ B_1(x_1),\ B_2(x_1))\ P\ (B_0(x_2),\ B_1(x_2),\ B_2(x_2))^t$$

with the 3×3 coefficient matrix $P = (p_{ij})$ with respect to the tensorial Bernstein basis. Moreover,

$$(B_0(x_1),\ B_1(x_1),\ B_2(x_1)) = (1,\ x_1,\ x_1^2)\ T$$

and

$$T = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

With $C = TPT^t$, it follows

$$\begin{aligned} c_{12} &= (-2p_{00} + 2p_{10}) - 2(-2p_{01} + 2p_{11}) + (-2p_{02} + 2p_{12}) = 0 \\ c_{21} &= -2(p_{00} - 2p_{10} + p_{20}) + 2(p_{01} - 2p_{11} + p_{21}) = 0 \\ c_{22} &= (p_{20} - 2p_{10} + p_{20}) - 2(p_{01} - 2p_{11} + p_{21}) \\ &\quad + (p_{02} - 2p_{12} + p_{22}) = 0 \end{aligned}$$

These equations concern the p_{ij}, *i.e.,* the coefficients of the linear objective function $\sum_{ij} \lambda_{ij} p_{ij}$, and not the polytope $\mathcal{B}_2$. Thus they do not seem to be helpful. For an illustration, consider the polytope $\mathcal{B}_2$ in three dimensions or beyond. Then the gradient of the linear objective function to be minimized or maximized must be horizontal, *i.e.,* only the first two coefficients are non-zero. This would be interesting if we would be able to compute the projection of the polytope $\mathcal{B}_2$ on the horizontal plane.

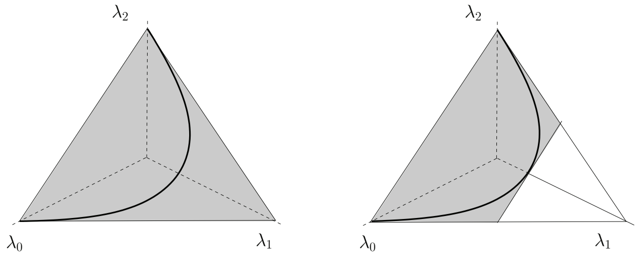

Figure 1: Left: The patch $(B_0(x), B_1(x), B_2(x)), x \in [0,1]$ is a curve. It is enclosed in a triangle $(\lambda_0, \lambda_1, \lambda_2)$ with $0 \le \lambda_i$ and $\lambda_0 + \lambda_1 + \lambda_2 = 1$. Right: The polytope is truncated with the inequality constraint $\lambda_1 \le 1/2$.

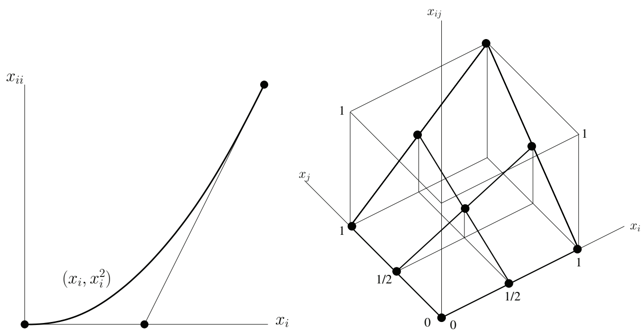

Figure 2: Left: Three control points of the parabola $(x_i, x_{ii} = x_i^2)$. Right: Three-by-three control points of the patch $(x_i, x_j, x_{ij} = x_i x_j)$.

4. SHRINKED CONVEX HULL RANGE

The second method for computing ranges of polynomials uses a truncated variant of the previous simplex. We call it the *shrinked convex hull.* After shrinking, the polytope is not a simplex anymore.

For $n = 1$, *i.e.,* one variable $\mathbf{x} = (x_1)$: $p(x_1) = p_0 B_0(x_1) + p_1 B_1(x_1) + p_2 B_2(x_1)$, and the simplex is the set of points $(\lambda_0, \lambda_1, \lambda_2)$, where all λ_i are non-negative and their sum equals 1. This simplex encloses the patch $(B_0(x_1),\ B_1(x_1),\ B_2(x_1))$ where $x_1 \in [0,1]$. Note that $B_1(x_1) \le 1/2$ so $\lambda_1 \le 1/2$. The previous simplex $(\lambda_0, \lambda_1, \lambda_2)$, which is a triangle, can be shrinked to a quadrilateral (Figure 1, right).

For $n = 2$ and $\mathbf{x} = (x_1, x_2)$ it is $B_1(x_1)(B_0(x_2) + B_1(x_2) + B_2(x_2)) = B_1(x_1) \le 1/2$. Thus we can add the linear inequality constraint $\lambda_{10} + \lambda_{11} + \lambda_{12} \le 1/2$. Similarly, $(B_0(x_1) + B_1(x_1) + B_2(x_1))B_1(x_2) = B_1(x_2) \le 1/2$ can be expressed as $\lambda_{01} + \lambda_{11} + \lambda_{21} \le 1/2$. Finally, $B_1(x_1) \le 1/2, B_1(x_2) \le 1/2 \Rightarrow B_1(x_1)B_1(x_2) \le 1/4$, thus we can add the linear inequality constraint $\lambda_{11} \le 1/4$. These three linear inequality constraints truncate the initial simplex and thus tighten the range bounds of polynomials.

For $n = 3$ and $\mathbf{x} = (x_1, x_2, x_3)$, we can shrink the simplex with seven linear inequality constraints: $\sum_i \sum_j \lambda_{1ij} \le 1/2$, $\sum_i \sum_j \lambda_{i1j} \le 1/2$, $\sum_i \sum_j \lambda_{ij1} \le 1/2$, $\sum_i \lambda_{11i} \le 1/4$, $\sum_i \lambda_{1i1} \le 1/4$, $\sum_i \lambda_{i11} \le 1/4$, and $\lambda_{111} \le 1/8$.

The generalization for arbitrary n is simple. It generates $2^n - 1$ linear inequality constraints. The volume of the polytope decreases, which narrows the range bounds of polynomials. The size of the linear programming problems is still exponential in the number n of variables.

5. ENCLOSING $\mathcal{Q}_N$

The quadratic patch $\mathcal{Q}_n$ is enclosed in the convex hull of its control points. We denote the problem variables as follows, x_{ii} for x_i^2, and x_{ij} for $x_i x_j$ $(1 \le i < j \le n)$. The representation of $\mathcal{Q}_n$ has 3^n control points: The control points for x_i are 0, 1/2, or 1. The control points for x_{ii} are 0 for $x_i = 0$ and $x_i = 1/2$, and 1 for $x_i = 1$. Figure 2 shows the control points for the parabola $(x_i, x_{ii} = x_i^2)$. The patch $(x_i, x_j, x_{ij} = x_i x_j)$ can be represented as the tensor product of three line segments. Figure 2 shows the nine control points of the patch $(x_i, x_j, x_{ij} = x_i x_j)$.

For $n = 2$, the $3^2 = 9$ control points $\mathbf{q}_i$ are

	x_1	x_2	x_{11}	x_{22}	x_{12}
$\mathbf{q}_1$	0	0	0	0	0
$\mathbf{q}_2$	0	1/2	0	0	0
$\mathbf{q}_3$	0	1	0	1	0
$\mathbf{q}_4$	1/2	0	0	0	0
$\mathbf{q}_5$	1/2	1/2	0	0	1/4
$\mathbf{q}_6$	1/2	1	0	1	1/2
$\mathbf{q}_7$	1	0	1	0	0
$\mathbf{q}_8$	1	1/2	1	0	1/2
$\mathbf{q}_9$	1	1	1	1	1

We compute properties of these polytopes for $n \le 4$.

	$n =$ 1	2	3	4
nb. coordinates	3	6	10	15
nb. vertices	3	$9 = 3^2$	$27 = 3^3$	$81 = 3^4$
nb. hyperfaces	3	18	173	46068
volume	$\frac{1}{4}$ = 0.25	$\frac{1}{96}$ = $1.041\overline{6}$e-2	$\frac{47}{645120}$ ≈ 7.285e-5	$\frac{375533}{4637432217600}$ ≈ 8.098e-8

The convex hull polytope of the 3^n control points $\mathbf{q}_i$ of $\mathcal{Q}_n$ is defined by 3^n variables λ_i, $i \in [1:3^n]$. It is the set of points $\sum_{i=1}^{3^n} \lambda_i \mathbf{q}_i$, where all λ_i are non-negative, and their sum equals 1.

For the range of

$$p(\mathbf{x}) = \sum_i a_i x_i^2 + \sum_{\{(i,j)|i<j\}} b_{ij} x_i x_j + \sum_i c_i x_i + d,$$

we determine the vertex of the previous polytope (the convex hull of the control points of $\mathcal{Q}_n$), which minimizes and maximizes the corresponding linear objective function

$$\sum_i a_i x_{ii} + \sum_{\{(i,j)|i<j\}} b_{ij} x_{ij} + \sum_i c_i x_i + d$$

with variables λ_i, $(x_1, \ldots, x_n, x_{11}, \ldots x_{nn}, x_{12}, \ldots, x_{n-1,n}) = \sum_{i=1}^{3^n} \lambda_i \mathbf{q}_i$ and constraints $0 \le \lambda_i$, $\sum_{i=1}^{3^n} \lambda_i = 1$.

We do not discuss shrinking this convex hull as it still incorporates an exponential number n of variables. But the polytope inspires the practicable polytope of the next section.

We give the number of hyperfaces of these polytopes for $n \le 4$, computed with the program *lrs* by D. Avis. Of course, the number of hyperfaces is always greater than 3^n.

6. BERNSTEIN POLYTOPE

The Bernstein polytope $\mathcal{P}_n$ was introduced in [4]. The quadratic patch $\mathcal{Q}_n$ is enclosed in a polytope with a quadratic number of hyperplanes.

The projection of the patch $\mathcal{Q}_n$ to the plane (x_i, x_{ii}) is enclosed in a triangle (or in a quadrilateral), see Figure 3. The projection of $\mathcal{Q}_n$ to the subspace (x_i, x_j, x_{ij}) is enclosed in a tetrahedron (Figure 4). This tetrahedron is optimal, *i.e.,* it can not be shrunk. It is the convex hull of the projection of $\mathcal{Q}_n$ to the subspace (x_i, x_j, x_{ij}).

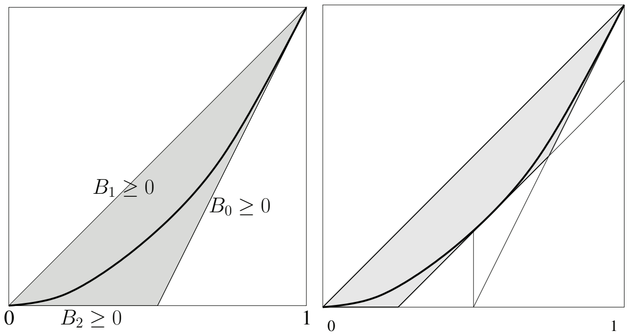

Figure 3: Left: The Bernstein polytope encloses the curve $(x, y = x^2)$ for $(x, y) \in [0,1]^2$. Its bounding halfspaces are $B_0(x) = (1-x)^2 = y - 2x + 1 \ge 0$, $B_1(x) = 2x(1-x) = 2x - 2y \ge 0$, $B_2(x) = x^2 = y \ge 0$. Right: A fourth constraint $(x - 1/2)^2 = x^2 - x + 1/4 \ge 0 \to y - x + 1/4 \ge 0$.

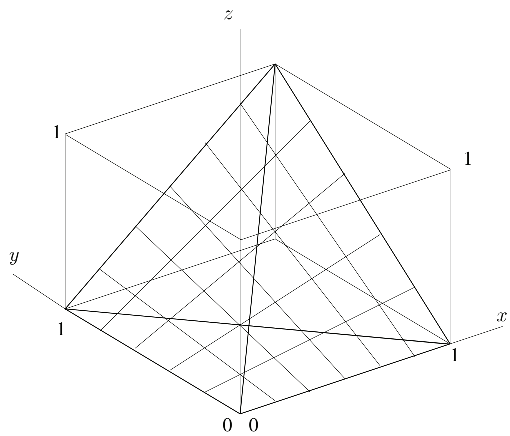

Figure 4: The Bernstein polytope enclosing the surface patch $(x, y, z = xy)$. Then the inequalities of bounding halfspaces are linearizations of $B_i^{(1)}(x) B_j^{(1)}(y) \ge 0$ with $i = 0, 1$, $j = 0, 1$. For instance, $B_0^{(1)}(x) B_0^{(1)}(y) = (1-x)(1-y) = 1 - x - y + xy \ge 0 \to 1 - x - y + z \ge 0$.

For small $n < 5$, *lrs* can explicitly compute the Bernstein polytope. The following table gives the projective dimension (*i.e.,* the number of homogeneous coordinates, subtract one for the affine dimension), the number of hyperplanes, the number of vertices, and the volume. An entry "?" denotes that the program was stopped after considerable time.

	$n =$ 1	2	3	4
$\mathcal{P}_n$				
nb. coordinates	3	6	10	15
nb. hyperfaces	3	10	21	36
nb. vertices	3	14	116	1688
volume	$\frac{1}{4}$ = 0.25	$\frac{1}{60}$ = $0.01\overline{6}$	$\frac{1}{2688}$ ≈ 0.000372	?

	$n=$ 1	2	3	4
$\mathcal{P}'_n$				
nb. coordinates	3	6	10	15
nb. hyperfaces	4	12	24	40
nb. vertices	4	28	464	17744
volume	$\frac{3}{16}$ = 0.1875	$\frac{1}{120}$ = $0.008\overline{3}$	$\frac{389}{3440640}$ ≈ 0.000113	?
$\mathcal{P}''_n$				
nb. coordinates	3	6	10	15
nb. hyperfaces	4	13	27	46
nb. vertices	4	26	525	42307
volume	$\frac{3}{16}$ = 0.1875	$\frac{109}{15360}$ ≈ 0.007096	?	?

It is possible to shrink this polytope further by adding linear inequality constraints. For instance, the triangle in the plane (x_i, x_{ii}) can be shrunk further by the inequality $(x_i - 1/2)^2 = x_i^2 - x_i + 1/4 \geq 0 \rightarrow x_{ii} - x_i + 1/4 \geq 0$ into a quadrilateral. We denote the resulting polytope by $\mathcal{P}'_n$.

Similarly in subspaces (x_i, x_j, x_{ij}), $(x_i - x_j)^2 \geq 0 \rightarrow x_{ii} - 2x_{ij} + x_{jj} \geq 0$ for $i < j$ provides a non-redundant inequality, which truncates the Bernstein polytope. We call the resulting polytope $\mathcal{P}''_n$.

7. EMPIRICAL COMPARISON

We measure the average width of the ranges of 100 random polynomials of total degree 2 by linear programming on each of the polytopes and by an interval arithmetic computation.

As random polynomials, we consider two classes of polynomials. For general polynomials, we choose all coefficients a_i, b_{ij}, c_i, d equally random from the interval $[-100, 100]$ in $p(\mathbf{x}) = \sum_i a_i x_i^2 + \sum_{\{(i,j)|i<j\}} b_{ij} x_i x_j + \sum_i c_i x_i + d$. Note that polynomials, encountered in practice, are often sparse, *i.e.,* most of the $\Theta(n^2)$ coefficients are zero.

In interval Newton solvers, preconditioning yields to special polynomials for the Newton map $N(\mathbf{x}) = \mathbf{x} - P(\mathbf{x})(P'(\mathbf{m}))^{-1}$. At the domain center $m_i = 1/2$, $i = 1, \ldots, n$, all derivatives of the polynomials in $N(\mathbf{x})$ vanish. In this class of polynomials, we choose polynomials $p(\mathbf{x}) = \sum_i a_i (x_i - 1/2)^2 + \sum_{\{(i,j)|i<j\}} b_{ij} (x_i - 1/2)(x_j - 1/2) + \sum_i c_i (x_i - 1/2) + d$ with coefficients a_i, b_{ij}, c_i, d equally random from the interval $[-100, 100]$.

Figures 5 and 6 give the average width of the polynomial ranges with its standard deviation as a function of the number n of variables. For small n, the differences between range widths of all methods are small. For large n, the range widths provided by the Bernstein polytope $\mathcal{P}_n$ (or one of its variants) are significantly better than the ones provided by the standard, centered form evaluation with interval arithmetic. The *standard, centered form* of a function $p(\mathbf{x})$, $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^n$ is given by

$$p(\mathbf{m}) + [p'](\mathcal{D}) \cdot (\mathcal{D} - \mathbf{m}) \text{ with the domain center } \mathbf{m} \in \mathcal{D}$$

For notation, we use the interval closure operator $[f]$ for an expression f in n variables $x_1, \ldots, x_n$, which can be defined inductively by the interval evaluation of f. Note that for a quadratic polynomial p, the derivative expression p' is linear, and $[p'](\mathcal{D})$ encloses its range tightly [7].

In comparison to the minimum/maximum-bounds of the TBB coefficients, the range widths provided by the Bernstein polytopes are only slightly larger (15% for $n = 10$). Among the TBB polytopes, the differences are very small, and could not be detected at all for the zero center-derivatives polynomials.

In the Bernstein polytope $\mathcal{P}'_n$ (Figure 3, right) (the triangles are replaced with quadrilaterals), the widths' improvements are not large, the average widths for $n = 10$ decreased by 6% only. The running times are also very similar so that the small gain is doubtful. Furthermore, in the Bernstein polytope $\mathcal{P}''_n$ (Figure 4), the average widths for $n = 10$ decreased by 11% but the polytope's number of hyperplanes roughly doubled. The resulting running time is more than doubled. Among the Bernstein polytopes, the inequalities $(x_i - x_j)^2 \geq 0$ are not worth the cost.

8. CONCLUSION

In this paper, we presented three different polytopes bounding coefficients with respect to the TBB. The TBB has an exponential size in the number n of variables. All three polytopes inherit the exponential size so that all methods for computing ranges based on them are exponential time as well. Furthermore, we derive a Bernstein polytope which is defined by a polynomial number of hyperplanes (in the number n of variables). We computed the number of vertices, the number of hyperplanes, and the volume of all polytopes as far as possible for small n. An empirical comparison of the average range widths for random polynomials computed using these polytopes shows that the Bernstein polytope provides only slightly worse bounds for $n \leq 10$. Although this difference will increase with even larger n, the method using the Bernstein polytope is the only one with acceptable runtime in practice. For using it, a linear program solver using a simplex method or an interior-point method is required.

9. REFERENCES

[1] O. BEAUMONT. PhD thesis *Algorithmique pour les intervalles.* Univ. Rennes 1, France, January 1999.

[2] A. DEZA, E. NEMATOLLAHI, AND T. TERLAKY. *How good are interior point methods? Klee-Minty cubes tighten iteration-complexity bounds.* Mathematical Programming 113 (2008), pp. 1–14.

[3] G. FARIN. *Curves and Surfaces for CAGD: A Practical Guide*. Academic Press Professional Inc, San Diego, CA, 1988.

[4] C. FUENFZIG, D. MICHELUCCI, AND S. FOUFOU. *Nonlinear Systems Solver in Floating Point Arithmetic Using LP reduction. ACM Symp. Solid and Physical Modeling 2009*, San Francisco, CA, pp. 123–134.

[5] A. A. GAGANOV. *Computational Complexity of the Range of the Polynomial in several variables* In *Cybernetics*, 1985, pp. 418–425.

[6] J. GARLOFF, AND A. P. SMITH. *Investigation of a subdivision based algorithm for solving systems of polynomial equations. Journal of nonlinear analysis : Series A Theory and Methods 47*, 1 (2001), pp. 167–178.

[7] L. JAULIN, M. KIEFFER, O. DIDRIT, AND E. WALTER. *Applied Interval Analysis. Springer*, 2001.

[8] R. MARTIN, H. SHOU, I. VOICULESCU, A. BOWYER, AND G. WANG. *Comparison of Interval Methods for Plotting Algebraic Curves*. Computer Aided Geometric Design 19(7) (2002), pp. 553–587.

[9] A. SCHRIJVER. *Theory of Linear and Integer Programming.* J. Wiley and Sons, 1986.

[10] E. C. SHERBROOKE, AND N. M. PATRIKALAKIS. Computation of the solutions of nonlinear polynomial systems. *Comput. Aided Geom. Des. 10*, 5 (1993), pp. 379–405.

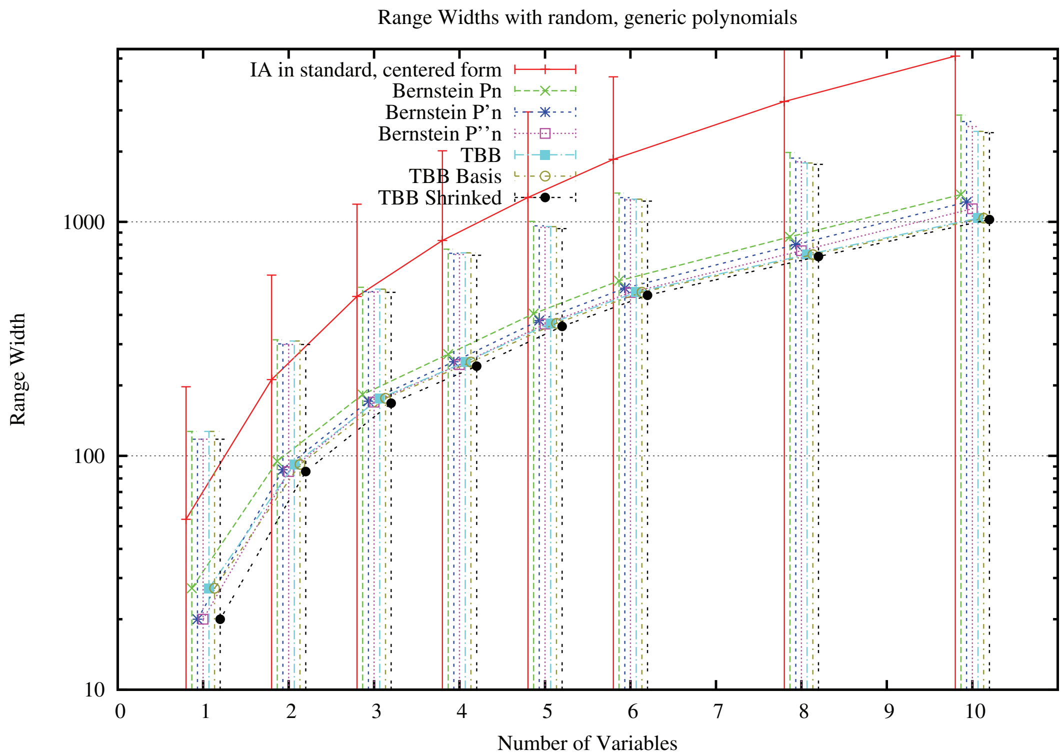

Figure 5: Statistics of range widths for random, quadratic polynomials. The graph shows the average widths with its standard deviations.

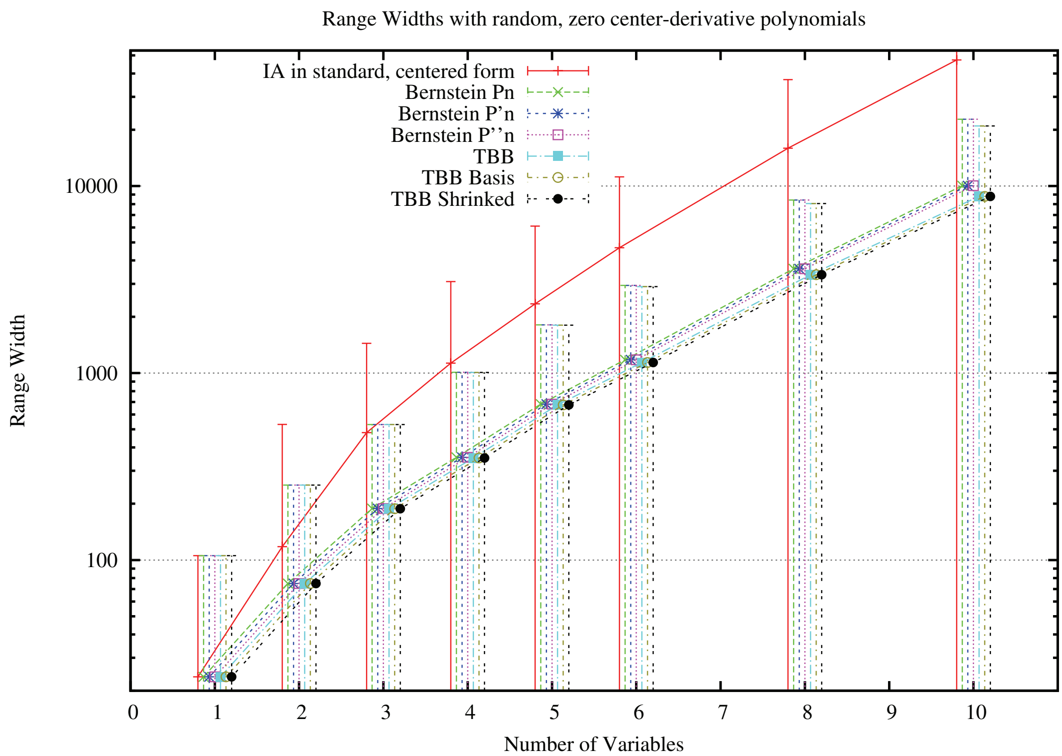

Figure 6: Statistics of range widths for random, quadratic polynomials with zero center-derivatives. The graph shows the average widths with its standard deviations.