Interval Computations
No 2, 1993

The Bernstein Algorithm

Jürgen Garloff

We solve the problem of finding an enclosure for the range of a multivariate polynomial over a rectangular region by expanding the given polynomial into Bernstein polynomials. Then the coefficients of the expansion provide lower and upper bounds for the range and these bounds converge monotonically if the degree of the Bernstein polynomials is elevated. To obtain a faster improvement of the bounds we use subdivision and present an economical procedure for computing the bounds on subboxes. Then we apply the results to a problem of robust control, viz. checking the (Hurwitz) stability of a polynomial with coefficients depending polynomially on parameters varying inside given intervals. Numerical examples are also presented.

Алгоритм Бернштейна

Ю. Гарлофф

Решается задача нахождения оболочки для множества значений многочлена многих переменных на прямоугольной области путем разложения заданного многочлена в многочлены Бернштейна. При этом коэффициенты разложения обеспечивают нижнюю и верхнюю границы множества значений, которые монотонно сходятся при увеличении степени многочленов Бернштейна. Чтобы ускорить сходимость границ, используется деление исходной области и вводится экономичная процедура вычисления границ прямоугольных областей, полученных делением. Результаты применяются в задаче робастного управления, а именно для проверки (гурвицевой) устойчивости многочлена, коэффициенты которого полиномиально зависят от параметров, изменяющихся внутри заданных интервалов. Представлены численные примеры.

Introduction

The Bernstein algorithm, e.g. [1, 2] and the references therein, is now a well established tool for computing bounds for the range of a multivariate polynomial over a rectangular region (for the univariate case cf. [3, 4]). Other approaches tailored for this class of functions include the use of centered forms [5] and sampling the polynomial at certain points [1, 6]. Methods for enclosing the range of more general functions may be found in [7]. Compared with these methods the Bernstein algorithm has the advantage that it avoids function evaluations which might be costly if the degree of the polynomial is high. A salient feature of the Bernstein algorithm is that the computation of the bounds conveys an information about the sharpness of these bounds. This knowledge plays an important role in speeding up the convergence of the algorithm. A disadvantage is that the approach is presently restricted to polynomials. The approach is not restricted to rectangular regions since triangular regions can be handled in a similar way, cf. [1].

Once bounds for the range of a multivariate polynomial are computed by the Bernstein algorithm, these bounds may be improved, e.g. by elevation of the degree of the Bernstein polynomials or by subdivision. Bernstein subdivision techniques have proved to be profitable in a wide variety of algorithms for computer aided geometric design, e.g. for algebraic curve intersection [8], intersection of a ray with a trimmed rational Bézier surface patch [9], computation of all solutions of a system of a nonlinear polynomial equations which lie within a box [10], and computation of the singularities and intersections of offsets of planar integral polynomial curves [11]. Since it turns out that in the above range problem degree evaluation is inferior to subdivision[1] we concentrate here on subdivision techniques and present a method which leads to a considerable saving of computational cost.

The organization of the paper is as follows: in the next section we describe the method for enclosing the range of a multivariate polynomial over the unit box. In Section 1.1 we recall the basic properties of the Bernstein coefficients. In Section 1.2 we apply subdivision and show how the Bernstein coefficients of the polynomial on subboxes can be computed with less computational effort.

In Section 2 we apply our results to a robust control problem, viz. check-

[1]Farouki and Rajan [12] report similar experiences in their investigation on the root condition of polynomials in Bernstein form.

ing the (Hurwitz) stability of a polynomial with coefficients depending polynomially on parameters varying inside given intervals. For more general problems in control theory which may be solved by the Bernstein algorithm see [2]. Applications in other fields include e.g. testing the hypothesis in global univalence theorems of Gale-Nikaido type, cf. [13], if the function to be checked for univalence is in each component a multivariate polynomial.

Numerical examples are given in the last section.

1 Bounds for a multivariate polynomial over the unit box

1.1 Basic properties of Bernstein coefficients

In this section we expand a given multivariate polynomial into Bernstein polynomials to obtain bounds for the range of this polynomial over the *unit box* I which is a Cartesian power of the unit interval $[0, 1]$. That we consider here the unit box I is no restriction since any nonempty rectangular region, i.e. the Cartesian product of nonempty compact real intervals, can be mapped affinely onto I. Furthermore, we shall present the results only for the minimum of the range since the results for the maximum are completely analogous.

Let p be a polynomial (of degree r) in the variables $x_1, \ldots, x_q$

$$p(x) = \sum_{i_1,\ldots,i_q=0}^{r} a_{i_1\ldots i_q} \prod_{j=1}^{q} x_j^{i_j} \tag{1.1}$$

where all coefficients $a_{i_1\ldots i_q}$ are real and $x = (x_1, \ldots, x_q) \in \mathbb{R}^q$. We want to know

$$\underline{m} = \min_{x\in I} p(x).$$

To shorten the following presentation we shall often write i instead of $i_1 i_2 \ldots i_q$ with range

$$K = \big\{(i_1, i_2, \ldots, i_q) \mid i_1, \ldots, i_q = 0, 1, \ldots, k\big\}.$$

The *Bernstein polynomials* of degree k are defined by

$$p_i^{(k)}(x) = \prod_{j=1}^{q} \binom{k}{i_j} x_j^{i_j} (1 - x_j)^{k-i_j}, \quad i = (i_1 \dots i_q) \in K.$$

Expansion of the given polynomial p in the power form (1.1) into Bernstein polynomials results[2] in ($k \geq r$)

$$p(x) = \sum_{i \in K} b_i^{(k)} p_i^{(k)}(x), \quad x \in I$$

where the *Bernstein coefficients* $b_i^{(k)}$ are given for $(i_1 \dots i_q) \in K$ by

$$b_i^{(k)} = b_{i_1 \dots i_q}^{(k)} = \sum_{\ell_1=0}^{i_1} \cdots \sum_{\ell_q=0}^{i_q} \prod_{j=1}^{q} \binom{i_j}{\ell_j} \rho_{\ell_1 \dots \ell_q}^{(k)} a_{\ell_1 \dots \ell_q}$$

with $\rho_{\ell_1 \dots \ell_q}^{(k)} = \left[\prod_{j=1}^{q} \binom{k}{\ell_j}\right]^{-1}$, and $a_{\ell_1 \dots \ell_q} = 0$ if some ℓ_j is such that $\ell_j > r$.

We set

$$\underline{\beta}^{(k)} = \min_{i \in K} b_i^{(k)}.$$

Theorem 1 [1, 2]. *For each $k \geq r$ we have*

(i) $$\underline{\beta}^{(k)} \leq \underline{m} \leq \underline{\beta}^{(k)} + \gamma(k-1)k^{-2}$$

where $\gamma = \sum_{i_1, \dots, i_q=0}^{r} \sum_{\substack{j=1 \\ i_j \neq 0}}^{q} (i_j - 1)^2 |a_{i_1 \dots i_q}|$;

(ii) $\underline{\beta}^{(k)} = \underline{m}$ *iff* $\underline{\beta}^{(k)} = b_{i_1 \dots i_q}^{(k)}$ *with* $i_j \in (0, k)$, $j = 1, 2, \dots, q$.

Furthermore, it can be shown that the sequence of bounds $\left\{\underline{\beta}^{(k)}\right\}$ converges monotonically [1].

Note that if condition (ii) of Theorem 1 holds, then p assumes its minimum at a vertex of I. Therefore, condition (ii) will be referred to as the *vertex condition*. However, p may achieve its minimum at a vertex of I while

[2]The question of stability of the transformation between the power and the Bernstein polynomial forms was addressed in [14, 15].

the vertex condition does not hold.

Example 1. Let p be the Chebyshev polynomial of 10th degree, i.e.

$$p(x) = 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1.$$

It is well known that the minimum of p on $[0,1]$ is -1. Therefore, we have

$$\underline{m} = p(0) = b_0^{(10)}$$

whereas

$$\underline{\beta}^{(10)} = b_9^{(10)} = -9.$$

In [1] a difference table method for computing the $(k+1)^q$ Bernstein coefficients is described. The number of operations required by this method is $qk(k+1)^q/2$ additions (multiplications with ρ_i neglected).

1.2 Subdivision

If we want to improve the bounds (if e.g. the vertex condition does not hold) we may elevate the degree k of the Bernstein polynomials. But numerical examples [16] show that the convergence of the sequence $\left\{\underline{\beta}^{(k)}\right\}$ is rather slow (cf. the upper bound in Theorem 1 (i)). A better way to get tighter bounds is to apply subdivision.

We divide the unit box I into 2^q subboxes of edge length 1/2 and calculate the Bernstein coefficients of p on these subboxes, i.e. the Bernstein coefficients of the 2^q polynomials obtained when p is shifted from the subboxes to I. The process may be continued by subdividing again each of the 2^q subboxes into 2^q subboxes of edge length 1/4 and calculating the Bernstein coefficients on all resulting subboxes and so on. Then the minimum of the Bernstein coefficients of p on all subboxes at a fixed subdivision level s, where $s = 0$ refers to I, will be denoted by $\underline{\beta}_{(s)}$. In the following the degree k of the Bernstein polynomials is fixed (usually one takes $k = r$) and therefore we suppress the upper index (k).

The following theorem shows that the sequence $\{\underline{\beta}_{(s)}\}$ converges to $\underline{m}$ quadratically with respect to the edge length of the subboxes generated by subdivision (for the univariate case cf. [4]).

Theorem 2 [2].

(i)
$$\underline{\beta}_{(s)} \leq \underline{m} \leq \underline{\beta}_{(s)} + \varepsilon 2^{-2s}$$

where ε is a constant not depending on s;

(ii)
$$\underline{\beta}_{(s)} = \underline{m} \quad \text{iff } \underline{\beta}_{(s)} \text{ fulfills a vertex condition on a subbox.}$$

The following proposition shows that the Bernstein coefficients at the first subdivision level can be computed from the Bernstein coefficients of p on I and therefore explicit transformation of the subboxes onto I is avoided. To facilitate the description, we introduce the following notation for the subboxes.

Let

$$I_{\nu_1 \ldots \nu_q} = X_{\nu_1} \times \cdots \times X_{\nu_q}$$

$$\nu = (\nu_1, \ldots, \nu_q) \in \{0, 1\}^q$$

where ν_j are Boolean variables and the intervals X_j are given by

$$X_{\nu_j} = \begin{cases} [0, 0.5] & \text{if } \nu_j = 0 \\ [0.5, 1] & \text{if } \nu_j = 1. \end{cases}$$

Let $b_i(\nu)$, $i \in K$, denote the Bernstein coefficients on the subbox I_ν.

Proposition 1 [1, 2]. For all $\nu = (\nu_1, \ldots, \nu_q) \in \{0, 1\}^q$ the following relations hold

$$b_{i_1 \ldots i_q}(\nu) = 2^{-\sum_{j=1}^{q} i_j} \sum_{s_1=0}^{i_1} \cdots \sum_{s_q=0}^{i_q} \prod_{j=1}^{q} \binom{i_j}{s_j} b^*_{s_1 \ldots s_q}(\nu)$$

where $b^*_{s_1 \ldots s_q}(\nu) = b_{t_1 \ldots t_q}$

$$\text{with} \quad t_j = \begin{cases} s_j & \text{if } \nu_j = 0 \\ k - s_j & \text{if } \nu_j = 1 \end{cases} \quad \text{for } j = 1, \ldots, q.$$

Based on Proposition 1 the computational effort for computing the Bernstein coefficients on subboxes can be brought down considerably by the following procedure:

Sweep in the first coordinate direction:

On $I_{0\ldots 0}$ (we suppress the explicit reference to this subbox):

Put $b_i^{(1,0)} := b_i, \quad i \in K$.

Then define for $\ell = 1, \ldots, k$

$$b_{i_1\ldots i_q}^{(1,\ell)} := \left\{ \begin{array}{ll} b_{i_1\ldots i_q}^{(1,\ell-1)}, & i_1 = 0, \ldots, \ell-1 \\ \frac{1}{2}\left(b_{i_1-1,i_2\ldots i_q}^{(1,\ell-1)} + b_{i_1\ldots i_q}^{(1,\ell-1)}\right), & i_1 = \ell, \ldots, k \end{array} \right\}$$

for $i_2, \ldots, i_q = 0, \ldots, k$.

We obtain as intermediate values of this computation the entries resulting by a sweep in the same coordinate direction on the neighbouring subbox $I_{10\ldots 0}$:

$$b_{i_1\ldots i_q}^{(1,k)}(10\ldots 0) = b_{ki_2\ldots i_q}^{(1,i_1)}(0\ldots 0), \quad (i_1, \ldots, i_q) \in K.$$

Sweep in m-th coordinate direction, $2 \leq m \leq q$:

For $\nu_1, \ldots, \nu_{m-1} = 0, 1$ (we suppress the explicit reference to $I_{\nu_1,\ldots,\nu_{m-1},0,\ldots,0}$ in the following formulae):

Put $b_i^{(m,0)} := b_i^{(m-1,k)}, \quad i \in K$.

Then define for $\ell = 1, \ldots, k$

$$b_{i_1\ldots i_q}^{(m,\ell)} := \left\{ \begin{array}{ll} b_{i_1\ldots i_q}^{(m,\ell-1)}, & i_m = 0, \ldots, \ell-1 \\ \frac{1}{2}\left(b_{i_1,\ldots,i_{m-1},i_m-1,i_{m+1},\ldots,i_q}^{(m,\ell-1)} + b_{i_1\ldots i_q}^{(m,\ell-1)}\right), & i_m = \ell, \ldots, k \end{array} \right\}$$

for $i_1, \ldots, \hat{\imath}_m, \ldots, i_q = 0, \ldots, k$.

These sweeps provide as byproducts the entries on the neighbouring subboxes $I_{\nu_1,\ldots,\nu_{m-1},1,0,\ldots,0}$, $\nu_1, \ldots, \nu_{m-1} = 0, 1$, resulting when the sweeps in the m first coordinate directions are performed on these subboxes, viz.

$$\begin{aligned} & b_{i_1\ldots i_q}^{(m,k)}(\nu_1, \ldots, \nu_{m-1}, 1, 0, \ldots, 0) \\ & \quad = b_{i_1,\ldots,i_{m-1},k,i_{m+1},\ldots,i_q}^{(m,i_m)}(\nu_1, \ldots, \nu_{m-1}, 0, \ldots, 0) \end{aligned}$$

with $(i_1, \ldots, i_q) \in K, \quad \nu_1, \ldots, \nu_{m-1} = 0, 1$.

After the last step rearrangement of the entries $b_i^{(q,k)}(\nu)$, $i \in K$, according to Proposition 1 gives the Bernstein coefficients of p on I_ν, $\nu \in \{0,1\}^q$.

The above procedure requires $k(k+1)^q(2^{q-1}-\frac{1}{2})$ additions and divisions by 2 (binary shifts) which is less than $1/q$ of the amount of work if the difference table method is applied to all 2^q subboxes.

The above procedure shows that the bound $\underline{\beta}_{(s)}$ is at least as good as the bound obtained without subdivision and that the bounds $\underline{\beta}_{(s)}$ converge monotonically as $s \to \infty$. Comparing with Theorem 1 (i), we see that subdivision is clearly superior to degree elevation. Furthermore, when refinement is applied iteratively the polynomial p will assume its minimum on a sufficiently small subbox at one of its vertices so that the chance is high that the bound provided by the Bernstein coefficients is sharp, cf. Theorem 2 (ii).

2 Application to a robust control problem

In this section we apply the results of Section 1 to the following robust control problem:

Let a polynomial ϕ

$$\phi(z) = a_0 z^m + a_1 z^{m-1} + \cdots + a_{m-1} z + a_m \tag{2.1}$$

be given with coefficients a_k depending polynomially on parameters $q_1, \ldots,$ $q_n, \quad q = (q_1, \ldots, q_n)$, i.e.

$$a_k(q) = \sum_{i_1,\ldots,i_n=0}^{r} a^{(k)}_{i_1,\ldots,i_n} q_1^{i_1} \cdots q_n^{i_n}$$

where the parameters vary inside given intervals $[\underline{q}_i, \overline{q}_i]$, $i = 1, \ldots, n$, i.e.

$$q \in Q = [\underline{q}_1, \overline{q}_1] \times [\underline{q}_2, \overline{q}_2] \times \cdots \times [\underline{q}_n, \overline{q}_n].$$

We assume $a_0(q) > 0$ for all $q \in Q$.

Then the following question arises:

Are the polynomials $\phi(q)$ (Hurwitz, asymptotically) stable for all $q \in Q$, i.e. are all zeros of $\phi(q)$ inside the open left half of the complex plane?

In the last years much attention was devoted to the solution of this problem, cf. the references in [2]. More recent approaches include [17, 18].

We use the following boundary crossing theorem[3] to transform the above problem into the problem of checking a multivariate polynomial for positivity over Q.

Boundary crossing theorem [21]

The polynomial (2.1) is stable for all $q \in Q$ iff:

1. *There exists a $q_0 \in Q$ for which $\phi(q_0)$ is stable;*
2. $\det H\big(\phi(q)\big) > 0$ *for all $q \in Q$, where $H(\phi) = \big(h_{i,j}(\phi)\big)$ is the Hurwitz matrix associated with ϕ, i.e.*

$$h_{i,j}(\phi) = a_{2j-i},\ i,\, j = 1,\, \ldots,\, m \quad \textit{with } a_\ell = 0 \quad \textit{for } \ell < 0 \textit{ or } \ell > m.$$

Note that the determinant of the Hurwitz matrix associated with the polynomial (2.1) is an n-variate polynomial, again named p (for larger degree p may be computed by a symbolic manipulation package).

After having transformed the given parameter set Q onto the unit box I, we calculate the minimum of the Bernstein coefficients of p on I. If this minimum is positive, $\phi(q)$ is stable for all $q \in Q$. Otherwise we subdivide I iteratively until positivity is achieved on all subboxes (then $\phi(q)$ is stable for all $q \in Q$ or the maximum[4] of the Bernstein coefficients is nonpositive on a subbox (then $\phi(q_0)$ is not stable for some $q_0 \in Q$).

Based on this procedure, a subdivision algorithm (called *breadth-first algorithm*) can be designed in the spirit of the Moore-Skelboe algorithm for unconstrained global optimization, e.g. [19]. Here the subboxes generated by subdivision do not have the same edge length step by step. In each iteration step subdivision is continued on a subbox which has smallest Bernstein coefficient since than the chances are best for finding a subbox on which p is not positive.

However, a serious drawback of this algorithm is that the number of arrays of the Bernstein coefficients which are generated in each subdivision

[3]This theorem allows that the coefficients a_k depend continuously on the parameters $q_1,\, \ldots,\, q_n$ (not necessarily polynomially). Therefore, also this more general problem may be transformed into a positivity check which may be treated efficiently by interval methods [7, 19, 20].

[4]As we have already noted at the beginning of Section 1.1, analogous results hold also for the maximum of the Bernstein coefficients.

step is in general equal to 2^n. To reduce the storage requirements we propose instead a *depth-first algorithm* which makes use of the relations between the Bernstein coefficients on neighbouring subboxes derived in Section 1.2.

In the sequel, all subboxes Y, $\widetilde{Y}$ are subboxes of I generated by subdivision and the array $b_i(Y)$ of a triple $\big(Y, t, b_i(Y)\big)$ denotes the Bernstein coefficients on $Y\,(t = n)$ or their intermediate values obtained after t sweeps in the first t coordinate directions $(t < n)$.

Depth-first subdivision algorithm

1. Calculate the Bernstein coefficients $b_i(I)$.
   Initialize list $\mathbf{L} = \big\{\big(I, n, b_i(I)\big)\big\}$.
2. Denote the first triple of $\mathbf{L}$ by $\big(Y, t, b_i(Y)\big)$.

   If $t < n$: Perform the sweeps on Y in the coordinate directions $t+1$, $t+2$, $\ldots$, n giving the Bernstein coefficients on Y named again $b_i(Y)$ replacing the intermediate values; replace t by n. Enter into $\mathbf{L}$ in the given order just behind $\big(Y, n, b_i(Y)\big)$ the subboxes $\big(\widetilde{Y}, j, b_i(\widetilde{Y})\big)$, $j = n, n-1, \ldots, t+1$, having an $(n-1)$-dimensional face in common with Y which is parallel to the $(x_1, x_2, \ldots, \hat{x}_j, \ldots, x_n)$-hyperplain, where $b_i(\widetilde{Y})$ are the (intermediate values of the) Bernstein coefficients on $\widetilde{Y}$ obtained as byproducts during the computation of the Bernstein coefficients on Y.

3. Calculate $a := \min b_i(Y)$.
   If ($a \leq 0$ and a is sharp) or $\max b_i(Y) \leq 0$ go to 7.
   If $a > 0$ go to 5.
4. Subdivide Y and replace in $\mathbf{L}$ the triple $\big(Y, n, b_i(Y)\big)$ by $\big(Y_{0\ldots0}, 0, b_i(Y)\big)$.
   Go to 2.
5. Remove $\big(Y, n, b_i(Y)\big)$ from list $\mathbf{L}$.
   If $\mathbf{L} = \emptyset$ go to 6.
   Go to 2.
6. Stop and report that p is positive on I.
7. Stop and report that p is not positive on I.

3 Computing the stability margin

In this section we present numerical examples for a problem closely related to the stability problem discussed in Section 2:

Let the polynomial $\phi(q^0)$ *be stable. We want to find the largest* ρ *named* ρ^* *such that all polynomials* $\phi(q)$ *are stable for all* q *with* $\|q - q^0\|_\infty^w < \rho$*, where* $\|\cdot\|_\infty^w$ *denotes the weighted infinity norm, i.e.* $\|q - q^0\|_\infty^w = \max_i w_i^{-1}|q_i - q_i^0|$, $w_i > 0$.

The quantity ρ^* is called the *stability margin*. The algorithm presented in this paper can be used to compute ρ^* by a bisection search over ρ involving a positivity check at each step.

The following examples were run (in double precision) on a workstation HP-9000/700 [22].

Example 3 [23]. Let

$$\phi(q_1, q_2; z) = z^3 + (q_1 + q_2 + 1)z^2 + (q_1 + q_2 + 3)z + 6q_1 + 6q_2 + 2q_1q_2 + 1.25.$$

The stability region in parameter space contains an instability disc centered around the point $(1, 1)$ with radius 0.5. The parameter vector q^0 is chosen as $q^0 = (1.6, 0.3)$, and the perturbation weight vector is $(0.15, 0.05)$. It can be shown that in this example $\rho^* = 4$. The polynomial to be checked for positivity is

$$\begin{aligned}\phi(q_1, q_2) \;=\; & 6q_1^3 + 6q_2^3 + 2q_1^2q_2 + 2q_1q_2^2 + 2q_1^3q_2 + 2q_1q_2^3 \\ & - 10.75q_1^2 - 10.75q_2^2 - 20.5q_1q_2 + 8q_1 + 8q_2 + 2.1875.\end{aligned}$$

Starting with $\rho = 1$, positivity was checked for (the respective deepest subdivision level is given in brackets) ρ =1 (0), 2 (0), 4 (0), 8 (2), 6 (3), 5 (4), 4.5 (5), 4.25 (5), 4.125 (6), 4.0625 (7), ... (successively halving) ... , 4.0009765625 (13). Since the required precision was reached, in a final step the last ρ was subtracted from the last but one giving $\rho = 4$. The program reported that the bound provided by the minimum of the Bernstein coefficients for $\rho = 4$ is sharp. The computing time for this bisection search was 120 ms. The breadth-first subdivision algorithm needed 150 ms with a maximum subdivision level of 7.

Example 4. The polynomial to be checked for positivity in the example presented in [24] is

$$\phi(q_1, q_2, q_3) = q_3^3\big(q_1^6 q_2^6 q_3^3 - q_1^6 q_2^2 q_3^3 - q_1^2 q_2^6 q_3^4\big) \qquad \text{with } q^0 = (1.4, 1.5, 0.8).$$

The perturbation weight vector is $w = (0.25, 0.2, 0.2)$. The stability margin was found to be between 1.0898... and 1.0908... . The computing time for the 13 bisection steps was 0.4 s. No subdivisions were needed.

Using the above factorization (which is quite natural since in the last column of the Hurwitz matrix only the (n, n)-entry is nonzero), the result was obtained in 0.24 s. It is our experience that only for higher degree polynomials it is advantageous to use this factorization.

Example 5. This example is taken from [25]:

$$\phi(z) = u(z)v(z) + x(z)y(z)$$

with

$$\begin{array}{rclcrcl} u(z) & = & r_1 z + r_2, & \qquad & x(z) & = & z^2 - r_3 z + r_4, \\ v(z) & = & q_1 z + q_2, & \qquad & y(z) & = & z^2 + q_3 z + q_4. \end{array}$$

The parameters r_i, $i = 1, 2, 3, 4$, belong to the following fixed intervals (not depending on ρ).

$$\begin{array}{rclcrcl} r_1 & \in & [2.7, 3.3], & \qquad & r_2 & \in & [1.7, 2.3], \\ r_3 & \in & [2.5, 3.5], & \qquad & r_4 & \in & [9.5, 10.5]. \end{array}$$

Wanted is the stability margin with respect to parameter vector $q = (q_1, q_2, q_3, q_4)$ assuming that $q^0 = (20, 23, 10, 5)$ and $w = (1, 1, 1, 1)$. The polynomial to be checked for positivity is a polynomial in 8 variables comprising 65 monomials. Positivity tests were performed for $\rho = 1$, 0.5, 0.25, 0.125, 0.1875, 0.15625, 0.171875, ..., 0.1865..., 0.1855.... The stability margin was found to be between the last two values of ρ. The computing time for these 12 bisection steps was 47.4 s and no subdivisions were needed.

Acknowledgements

We are grateful to Mrs. Dipl.-Ing. Margrit Schneider for productive discussions and to Mr. cand. inf. Siegmar Maier for his job of programming the algorithm.

References

[1] Garloff, J. *Convergent bounds for the range of multivariate polynomials.* Interval Mathematics (1985), Lecture Notes in Computer Science **212**, Springer, 1986, pp. 37–56.

[2] Malan, S., Milanese, M., Taragna, M., and Garloff, J. *B^3 algorithm for robust performances analysis in presence of mixed parametric and dynamic perturbations.* In: "Proceedings of the 31th IEEE Conference on Decision and Control, Tucson, Arisona, USA, 1992", IEEE Control System Society, 1992, pp. 128–133.

[3] Lane, J. M. and Riesenfeld, R. F. *Bounds on a polynomial.* BIT **21** (1981), pp. 112–117.

[4] Fischer, H. G. *Range computation and applications.* In: Ullrich, C. (ed.) "Contributions to computer arithmetic and self-validating numerical methods", SCAN'89, Baltzer, 1990, pp. 197–211.

[5] Qun Lin and Rokne, J. G. *A family of centered forms for a polynomial.* BIT **32** (1992), pp. 167–176.

[6] Gopalsamy, S., Khandekar, D., and Mudur, S. P. *A new method of evaluating compact geometric bounds for use in subdivision algorithms.* Computer Aided Geometric Design **8** (1991), pp. 337–356.

[7] Ratschek, H. and Rokne, J. *Computer methods for the range of functions.* Ellis Horwood, 1984.

[8] Sederberg, T. W. *An algorithm for algebraic curve intersection.* Computer Aided Design **21** (1989), pp. 547–554.

[9] Nishita, T., Sederberg, T. W., and Kakimoto, M. *Ray tracing trimmed rational surface patches.* Computer Graphics **24** (1990), pp. 337–345.

[10] Sherbrooke, E. C. and Patrikalakis, N. M. *Computation of the solutions of nonlinear polynomial systems.* Computer Aided Geometric Design **10** (1993), pp. 379–405.

[11] Maekawa, T. and Patrikalakis, N. M. *Computation of singularities and intersections of offsets of planar curves.* Computer Aided Geometric Design **10** (1993), pp. 407–429.

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

[13] Parthasarathy, T. *On global univalence theorems.* Lecture Notes in Mathematics **977**, Springer, 1983.

[14] Daniel, M. and Daubisse, J. C. *The numerical problem of using Bézier curves and surfaces in the power basis.* Computer Aided Geometric Design **6** (1989), pp. 121–128.

[15] Farouki, R. T. *On the stability of transformations between power and Bernstein polynomial forms.* Computer Aided Geometric Design **8** (1991), pp. 29–36.

[16] Zettler, M. *Gebietsunterteilung und graderhöhung bei Bernsteinpolynomen.* Diploma thesis, Fachhochschule Konstanz, 1991.

[17] Kaesbauer, D. *On robust stability of polynomials with polynomial parameter dependency: two/three parameter cases.* Automatica **29** (1993), pp. 215-217.

[18] Kolev, L. V. *A criterion for robust stability of polynomials with interval parameters.* Submitted to IEEE Trans. Circuits and Systems.

[19] Ratschek, H. and Rokne, J. *New computer methods for global optimization.* Ellis Horwood, 1988.

[20] Hansen, E. *Global optimization using interval analysis.* Marcel Dekker, 1992.

[21] Frazer, R. A. and Duncan, W. J. *On the criteria for the stability of small motions.* Proc. Royal Soc. A. **124** (1929), pp. 642–654.

[22] Maier, S. *Der Bernsteinalgorithmus mit Depth-first-Strategie.* Diploma thesis, Fachhochschule Konstanz, 1994.

[23] Ackermann, J., Hu, H. Z., and Kaesbauer, D. *Robustness analysis: a case study.* IEEE Trans. Automatic Control **35** (1990), pp. 352–356.

[24] Sideris, A. and Sánchez Peña, R. S. *Fast computation of the multivariable stability margin for real interrelated uncertain parameters.* In: “Proc. Amer. Control Conf.”, Atlanta, GA, 1988.

[25] Barmish, B. R. and Zhicheng Shi. *Robust stability of a class of polynomials with coefficients depending multilinearly on perturbations.* IEEE Trans. Automatic Control **35** (1990), pp. 1040–1043.

Fachbereich Informatik
Fachhochschule Konstanz
Postfach 100543
D–78405 Konstanz
FRG