Bernstein's basis and real root isolation

Bernard Mourrain, Fabrice Rouillier, Marie-Françoise Roy

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Bernstein's basis and real root isolation

Bernard Mourrain — Fabrice Rouillier — Marie-Françoise Roy

N° 5149

Mars 2004

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--5149--FR+ENG

Bernstein's basis and real root isolation

Bernard Mourrain* , Fabrice Rouillier † , Marie-Françoise Roy ‡

Thème 2 —Génie logiciel
et calcul symbolique
Projets GALAAD et SPACES

Rapport de recherche n° 5149 —Mars 2004 — 23 pages

Abstract: The Bernstein basis is widely used in Computer Aided Design. We explain how, combined with Descartes's rule, the Bernstein basis, provides as well an efficient method for real root isolation, using De Casteljau's algorithm, and make the link with more classical methods. Most of the content of the paper can be found in previous authors' articles. However, we present a new improved method for isolating real roots.

Key-words: Computer Algebra, Univariate Polynomials, Real Roots

* Projet GALAAD - INRIA - Bernard.Mourrain@inria.fr
† Projet SPACES - INRIA - Fabrice.Rouillier@inria.fr
‡ IRMAR- Université de Rennes 1 - Marie-Francoise.Roy@univ-rennes1.fr

Unité de recherche INRIA Rocquencourt

Bases de Bernstein et Isolation de Racines Réelles

Résumé : Les bases de Bernstein sont très utilisées en C.A.O. Nous montrons comment, combinées à la règle de Descartes, elles permettent de mettre au point des méthodes efficaces pour l'isolation des zéros réels de polynômes en une variable en utilisant l'algorithme de De Casteljau et nous établissons le lien avec des méthodes plus classiques. Une partie du contenu de cet article peut être trouvé dans certains articles auxquels nous avons contribué, mais nous proposons ici une nouvelle méthode.

Mots-clés : Calcul Formel, Polynômes en une variable, Zéros réels

1 Descartes's Law of Signs

The **number of sign changes**, $\mathrm{V}(a)$, in a sequence, $a = a_0, \cdots, a_p$, of elements in $\mathbb{R}\setminus\{0\}$ is defined by induction on p by:

$$\mathrm{V}(a_0) = 0$$
$$\mathrm{V}(a_0, \cdots, a_p) = \begin{cases} \mathrm{V}(a_1, \cdots, a_p) + 1 & \text{if } a_0a_1 < 0 \\ \mathrm{V}(a_1, \cdots, a_p) & \text{if } a_0a_1 > 0 \end{cases}$$

This definition extends to any finite sequence a of elements in $\mathbb{R}$ by considering the finite sequence b obtained by dropping the zeros in a and defining $\mathrm{V}(a) = \mathrm{V}(b)$, with the convention $\mathrm{V}(\emptyset) = 0$.

Let $\mathcal{P} = P_0, P_1, \ldots, P_d$ be a sequence of univariate polynomials, with coefficients in $\mathbb{R}$ and let a be an element of $\mathbb{R}\cup\{-\infty, +\infty\}$. The **number of sign changes** of $\mathcal{P}$ at a, denoted by $\mathrm{V}(\mathcal{P}; a)$, is $\mathrm{V}(P_0(a), \ldots, P_d(a))$ (at $-\infty$ and $+\infty$ the signs to consider are the signs of the leading monomials.

Given a and b in $\mathbb{R}\cup\{-\infty, +\infty\}$, we write $\mathrm{V}(\mathcal{P}; a, b)$ for $\mathrm{V}(\mathcal{P}; a) - \mathrm{V}(\mathcal{P}; b)$.

For example $\mathrm{V}(1, -1, 2, 0, 0, 3, 4, -5, -2, 0, 3) = 4$. If

$$\mathcal{P} = X^5, X^2 - 1, 0, X^2 - 1, X + 2, 1,$$

$\mathrm{V}(\mathcal{P}; 1) = 0$.

Let $P = a_pX^p + \cdots + a_0$ be a univariate polynomial in $\mathbb{R}[X]$. We write $\mathrm{V}(P)$ for the number of sign changes in $a_0, \ldots, a_p$ and $\mathrm{pos}(P)$ for the number of positive real roots of P, counted with multiplicity.

We state the famous Descartes's law of signs [2] (see for example [1] for a proof).

Theorem 1.1 (Descartes' law of signs)

$\mathrm{pos}(P) \leq \mathrm{V}(P)$,

$\mathrm{V}(P) - \mathrm{pos}(P)$ *is even.*

In general, it is not possible to conclude much about the number of roots on an interval using only Theorem 1.1.

An instance where Descartes's law of sign permits a sharp conclusion is the following.

Theorem 1.2 *Let*

$$\mathcal{D} = \{(x + iy) \in \mathbb{R}[i] \mid x < -\frac{1}{2}, (x + 1)^2 + y^2 < 1\}$$

be the part of the disk with center $(-1, 0)$ *and radius* 1 *which is to the left of the line* $x = -\frac{1}{2}$ *in* $\mathbb{R}^2 = \mathbb{R}[i]$. *If* $P \in \mathbb{R}[X]$ *is square free and has either no roots or exactly one simple root in* $(0, +\infty)$, *and all its complex roots in* $\mathcal{D}$, *then* $\mathrm{V}(P) = 0$ *or* $\mathrm{V}(P) = 1$ *and*

P *has one root in* $(0, +\infty)$ *if and only if* $\mathrm{V}(P) = 1$,

P has no root in $(0, +\infty)$ if and only if $\mathrm{V}(P) = 0$.

The proof of the theorem relies on the following lemmas.

Lemma 1.3 *For $A, B \in \mathbb{R}[X]$*

$$\mathrm{V}(A) = 0, \mathrm{V}(B) = 0 \Rightarrow \mathrm{V}(AB) = 0.$$

Proof: Obvious. □

Lemma 1.4 *For $A, B \in \mathbb{R}[X]$*

$$\mathrm{V}(A) = 1, B = X + b, b \geq 0 \Rightarrow \mathrm{V}(AB) = 1.$$

Proof: If $b = 0$, $\mathrm{V}(AB) = \mathrm{V}(A) = 1$. Now, let $b > 0$. Let

$$A = a_d X^d + a_{d-1} X^{d-1} + \ldots + a_0,$$

and suppose, without loss of generality, that $a_d = 1$. Since $\mathrm{V}(A) = 1$ and $a_d = 1$, there exists k such that

$$\begin{cases} a_i \geq 0 & \text{if } i > k, \\ a_k < 0, & \\ a_i \leq 0 & \text{if } i < k. \end{cases}$$

Letting c_i be the coefficient of X^i in AB and making the convention that $a_{d+1} = a_{-1} = 0$, we have

$$\begin{cases} c_i = a_{i-1} + a_i b \geq, 0 & \text{if } k+1 < i \leq d, \\ c_k = a_{k-1} + a_k b < 0, & \\ c_i = a_{i-1} + a_i b \leq 0, & \text{if } i < k, \end{cases}$$

and $c_{d+1} = a_d > 0$. So, whatever the sign of c_{k+1}, $\mathrm{V}(AB) = 1$. □

Lemma 1.5 *If $\mathrm{V}(A) = 1, B = X^2 + bX + c$ with $b > 1, b > c > 0$, then $\mathrm{V}(AB) = 1$.*

Proof: Let

$$A = a_d X^d + a_{d-1} X^{d-1} + \ldots + a_0,$$

and suppose without loss of generality that $a_d = 1$. Since $\mathrm{V}(P) = 1$ and $a_d = 1$, there exists k such that

$$\begin{cases} a_i \geq 0, & \text{if } i > k, \\ a_k < 0, & \\ a_i \leq 0, & \text{if } i < k. \end{cases}$$

Letting c_i be the coefficient of X^i in AB and making the convention that $a_{d+2} = a_{d+1} = a_{-1} = a_{-2} = 0$, we have

$$\begin{cases} c_i = a_{i-2} + a_{i-1} b + a_i c \geq 0, & \text{for } k+2 < i \leq d+2 \\ c_k = a_{k-2} + a_{k-1} b + a_k c < 0, & \\ c_i = a_{i-2} + a_{i-1} b + a_i c \leq 0, & \text{for } i < k. \end{cases}$$

The only way to have $\mathrm{V}(AB) > 1$ would be to have $c_{k+1} > 0, c_{k+2} < 0$, but this is impossible since

$$c_{k+2} - c_{k+1} = a_{k+2}c + a_{k+1}(b-c) + a_k(1-b) - a_{k-1} > 0.$$

□

Proof of Theorem 1.2: Notice first that by theorem 1.1,

$\mathrm{V}(P) = 1$ implies P has one root in $(0, +\infty)$ and

$\mathrm{V}(P) = 0$ implies P has no root in $(0, +\infty)$.

Note also that

if $X + a$ has its root in $(0, +\infty)$, then $a < 0$ and $\mathrm{V}(X + a) = 1$,

if $X + b$ has its root in $(-\infty, 0]$, then $b \geq 0$ and $\mathrm{V}(X + b) = 0$,

if $X^2 + bX + c$ has its roots in $\mathcal{D}$, then $b > 1, b > c > 0$ and $\mathrm{V}(X^2 + bX + c) = 0$.

Decompose now P into irreducible factors of degree 1 and 2 over $\mathbb{R}$. If P has one root a in $(0, +\infty)$, $\mathrm{V}(X + a) = 1$. Starting from $X + a$ and multiplying successively by the other irreducible factors of P, we get polynomials with sign variations equal to 1, using Lemma 1.4 and Lemma 1.5. Finally, $\mathrm{V}(P) = 1$.

If P has no root in $(0, +\infty)$, starting from 1 and multiplying successively by the irreducible factors of P, we get polynomials with sign variations equal to 0, using Lemma 1.3. Finally, $\mathrm{V}(P) = 0$. □

2 Bernstein's basis

The Bernstein's basis is widely used in Computer Aided Design [3]. We remind some of its important properties, in order to use them for real root isolation in the next section.

Notation 2.1 Let P be a polynomial of degree $\leq p$. The **Bernstein's polynomials** of degree p for c, d are the

$$B_{p,i}(c,d) = \binom{p}{i} \frac{(X-c)^{p-i}(d-X)^i}{(d-c)^p},$$

for $i = 0, \ldots, p$.

Remark 2.2 Note that $B_{p,i}(c,d) = B_{p,p-i}(d,c)$ and that

$$B_{p,i}(c,d) = \frac{(d-X)}{d-c} \frac{p}{i} B_{p-1,i-1}(c,d).$$

Since the valuation of the polynomials $B_{p,i}(c,d)$ at $x = c$ is i and $B_{p,i}(c,d)$ is a polynomial of degree p, we immediately deduce that $(B_{p,i}(c,d))$, $i = 0, \cdot, p$ form a basis of the vector space of polynomials of degree $\leq p$.

Here are some simple transformations, useful to understand the connection between Bernstein's basis and the monomial basis.

Reciprocal polynomial in degree p: $\mathrm{Rec}_p(P(X)) = X^p P(1/X)$. The non-zero roots of P are the inverses of the non-zero roots of $\mathrm{Rec}(P)$.

Contraction by ratio λ: for every non-zero λ, $\mathrm{C}_\lambda(P(X)) = P(\lambda X)$. The roots of $\mathrm{C}_\lambda(P)$ are of the form $\frac{x}{\lambda}$, where x is a root of P.

Translation by c: for every c, $\mathrm{T}_c(P(X)) = P(X - c)$. The roots of $\mathrm{T}_c(P(X))$ are of the form $x + c$ where x is a root of P.

Proposition 2.3 *Let $P = \sum_{i=0}^p b_i B_{p,p-i}(d,c) \in \mathbb{R}[X]$ be of degree $\leq p$. Let*

$$\mathrm{T}_{-1}\left(\mathrm{Rec}_p(\mathrm{C}_{d-c}(\mathrm{T}_{-c}(P)))\right) = \sum_{i=0}^{p} c_i X^i.$$

Then

$$\binom{p}{i} b_i = c_i.$$

Proof: Performing the contraction of ratio $d - c$ after translating by $-c$ transforms $\binom{p}{i}\frac{(X-c)^{p-i}(d-X)^i}{(d-c)^p}$ into $\binom{p}{i}X^{p-i}(1-X)^i$. Translating by -1 after taking the reciprocal polynomial in degree p transforms $\binom{p}{i}X^{p-i}(1-X)^i$ into $\binom{p}{i}X^i$. □

We denote as usual by $\mathrm{V}(b)$ the number of sign changes in a list b.

Proposition 2.4 *Let P be of degree p. We denote by $b = b_0, \ldots, b_p$ the coefficients of P in the Bernstein's basis of c, d. Let $n(P;(c,d))$ be the number of roots of P in (c,d) counted with multiplicities. Then*

$\mathrm{V}(b) \geq n(P;(c,d))$,

$\mathrm{V}(b) - n(P;(c,d))$ *is even.*

Proof: The claim follows immediately from Descartes's law of signs (Theorem 1.1), using Proposition 2.3. Indeed, the image of (c,d) under the translation by $-c$ followed by the contraction of ratio $d - c$ is $(0,1)$. The image of $(0,1)$ under the inversion $z \mapsto 1/z$ is $(1, +\infty)$. Finally, translating by -1 gives $(0, +\infty)$. □

Theorem 2.5 (Theorem of 2 circles) *If P is square free and has either no root or exactly one simple root in (c,d) and P has no complex root in $\mathcal{C}(c,d)_0 \cup \mathcal{C}(c,d)_1$, then*

P has one root in (c,d) if and only if $\mathrm{V}(b) = 1$,

P has no root in (c,d) if and only if $\mathrm{V}(b) = 0$.

Proof: We identify $\mathbb{R}^2$ with $\mathrm{C} = \mathbb{R}[i]$. The image of the complement of $\mathcal{C}(c,d)_0$ (resp $\mathcal{C}(c,d)_1$) under the translation by $-c$ followed by the contraction of ratio $d-c$ is the complement of $\mathcal{C}(0,1)_0$ (resp $\mathcal{C}(0,1)_1$). The image of the complement of $\mathcal{C}(0,1)_0$ under the inversion $z \mapsto 1/z$ is

$$\{(x+iy) \in \mathbb{R}[i] \mid 0 < x^2+y^2 < 1\}.$$

The image of the complement of $\mathcal{C}(0,1)_1$ under the inversion $z \mapsto 1/z$ is

$$\{(x+iy) \in \mathbb{R}[i] \mid x < \frac{1}{2}\}.$$

The image of the complement of $\mathcal{C}(0,1)_0 \cup \mathcal{C}(0,1)_1$ under the inversion $z \mapsto 1/z$ is

$$\{(x+iy) \in \mathbb{R}[i] \mid 0 < x^2+y^2 < 1, x < \frac{1}{2}\}.$$

Translating this region by -1, we get the region

$$\mathcal{D} = \{(x+iy) \mid x < -\frac{1}{2}, (x+1)^2+y^2 < 1\}$$

defined in Theorem 1.2.

The statement then follows from Theorem 1.2 and Proposition 2.3. □

The coefficients $b = b_0, \ldots, b_p$ of P in the Bernstein's basis of c, d give a rough idea of the shape of the polynomial P on the interval c, d. The **control polygon of P on** $[c,d]$ is the union of the segments $[M_i, M_{i+1}]$ for $i = 0, \ldots, p-1$, with

$$M_i = \left(\frac{ic+(p-i)d}{p}, b_i\right).$$

It is clear from the definitions that the graph of P goes through M_0 and M_p and that the line M_0, M_1 (resp M_{p-1}, M_p) is tangent to the graph of P at M_0 (resp. M_p).

Example 2.6 We take $p = 3$, and consider the polynomial P with coefficients $(4, -6, 7, 10)$ in the Bernstein's basis for $0, 1$

$$X^3, 3X^2(1-X), 3X(1-X)^2, (1-X)^3.$$

We draw the graph of P on $[0,1]$ the control line, and the X-axis in Figure 1 (see Annex).

The **control polygon of** P **on** $[c,d]$ is the convex hull of the points M_i for $i=1,\ldots,p$.

Example 2.7 Continuing Example 2.6, we draw the graph of P on $[0,1]$ and the control polygon in Figure 2 (see Annex).

An important and well-known property of the Bernstein polynomials is the following:

Proposition 2.8 *The graph of P on $[c,d]$ is contained in the convex hull of the control polygon of P on $[c,d]$.*

Indication of Proof of proposition 2.8: It is enough to prove that any line L above (resp. under) all the points in the control polygon of P on $[c,d]$ is above (resp. under) the graph of P on $[c,d]$. Use

$$\begin{aligned}
1 &= \sum_{i=0}^{p} \binom{p}{i} \left(\frac{X-c}{d-c}\right)^{p-i} \left(\frac{d-X}{d-c}\right)^{i} = \sum_{i=0}^{p} B_{p,i}(c,d) \\
X &= \left(d\left(\frac{X-c}{d-c}\right) + c\left(\frac{d-X}{d-c}\right)\right)\left(\frac{X-c}{d-c}+\frac{d-X}{d-c}\right)^{p-1} \\
&= \sum_{i=0}^{p-1} \left(d\left(\frac{X-c}{d-c}\right) + c\left(\frac{d-X}{d-c}\right)\right) B_{p-1,i}(c,d) \\
&= \sum_{i=0}^{p} \left(\frac{ic+(p-i)d}{p}\right) B_{p,i}(c,d).
\end{aligned}$$

□

The following algorithm, named De Casteljau, computes the coefficients of P in the Bernstein's bases of c,e and e,d from the coefficients of P in the Bernstein's basis of c,d.

Algorithm 2.9 (De Casteljau)

Input: *a list $b=b_0,\ldots,b_p$ representing a polynomial P of degree $\leq p$ in the Bernstein basis of c,d, and a number $e\in\mathbb{R}$.*

Output: *the list $b'=b'_0,\ldots,b'_p$ representing P in the Bernstein basis of c,ec and the list $b''=b''_0,\ldots,b''_p$ representing P in the Bernstein basis of e,d.*

Procedure:

Define $\alpha=\dfrac{e-c}{d-c}, \beta=\dfrac{d-e}{d-c}$.

Initialization: $b_j^{(0)} := b_j$, $j=0,\ldots,p$.

For $i=1,\ldots,p$,

For $j = 0, \ldots, p-i$, compute

$$b_j^{(i)} := \alpha b_j^{(i-1)} + \beta b_{j+1}^{(i-1)}$$

Define

$$b^{(p)} = b_0^{(0)}, \ldots, b_0^{(j)}, \ldots, b_0^{(p)}, \ldots, b_{p-j}^{(j)}, \ldots, b_p^{(0)},$$

and output

$$b' = b_0^{(0)}, \ldots, b_0^{(j)}, \ldots, b_0^{(p)}$$

and

$$b'' = b_0^{(p)}, \ldots, b_j^{(p-j)}, \ldots, b_p^{(0)}.$$

Algorithm 2.9 (De Casteljau) can be visualized with the following triangle.

$$\begin{array}{ccccccccccc}
b_0^{(0)} & & b_1^{(0)} & & \cdots & & \cdots & & b_{p-1}^{(0)} & & b_p^{(0)} \\
& b_0^{(1)} & & \cdots & & \cdots & & \cdots & & b_{p-1}^{(1)} & \\
& & \cdots & & \cdots & & \cdots & & \cdots & & \\
& & & \cdots & & \cdots & & \cdots & & & \\
& & & & b_0^{(p-1)} & & b_1^{(p)} & & & & \\
& & & & & b_0^{(p)} & & & & &
\end{array}$$

with $b_j^{(i)} := \alpha b_j^{(i-1)} + \beta b_{j+1}^{(i-1)}$, $\alpha = \dfrac{e-c}{d-c}, \beta = \dfrac{d-e}{d-c}$.

The coefficients of P in the Bernstein's basis of c, d appear in the top side of the triangle and the coefficients of P in the Bernstein's basis of c, e and e, d appear in the two other sides of the triangle.

For the sake of completeness, we give here the proof of completeness of Algorithm 2.9.

Proof of correctness of Algorithm 2.9: It is enough to prove the part of the claim concerning d, e. Indeed, by Remark 2.2, $\tilde{b}$ represents P in the Bernstein basis of d, c, and the claim is obtained by applying Algorithm 2.9 (De Casteljau) to $\tilde{b}$ at e. The output is $\tilde{b}''$ and $\tilde{b}$ and the conclusion follows using again Remark 2.2.

Let $\delta_{p,i}$ be the list of length $p+1$ consisting all zeroes except a 1 at the $i+1$-th place. Note that $\delta_{p,i}$ is the list of coefficients of $B_{p,i}(c, d)$ in the Bernstein's basis of c, d. We will prove that the coefficients of $B_{p,i}(c, d)$ in the Bernstein's basis of e, d coincide with the result of Algorithm 2.9 (De Casteljau) performed with input $\delta_{p,i}$. The correctness of Algorithm 2.9 (De Casteljau) for e, d then follows by linearity.

First notice that, since $\alpha = \dfrac{e-c}{d-c}, \beta = \dfrac{d-e}{d-c}$,

$$\frac{X-c}{d-c} = \alpha \frac{d-X}{d-e} + \beta \frac{X-e}{d-e},$$
$$\frac{d-X}{d-c} = \beta \frac{d-X}{d-e}.$$

Thus

$$\left(\frac{X-c}{d-c}\right)^{p-i} = \sum_{k=0}^{p-i} \binom{p-i}{k} \alpha^k \left(\frac{X-e}{d-e}\right)^{p-i-k} \left(\frac{d-X}{d-e}\right)^k,$$
$$\left(\frac{d-X}{d-c}\right)^{i} = \beta^i \left(\frac{d-X}{d-e}\right)^{i}.$$

It follows that

$$B_{p,i}(c,d) = \binom{p}{i} \sum_{j=i}^{p} \binom{p-i}{j-i} \alpha^{j-i} \beta^i \left(\frac{X-e}{d-e}\right)^{p-j} \left(\frac{d-X}{d-e}\right)^j.$$

Since

$$\binom{p}{i}\binom{p-i}{j-i} = \binom{j}{i}\binom{p}{j},$$
$$B_{p,i}(c,d) = \sum_{j=i}^{p} \binom{j}{i} \alpha^{j-i} \beta^i \binom{p}{j} \left(\frac{X-e}{d-e}\right)^{p-j} \left(\frac{d-X}{d-e}\right)^j.$$

Finally,

$$B_{p,i}(c,d) = \sum_{j=i}^{p} \binom{j}{i} \alpha^{j-i} \beta^i B_{p,j}(d,e).$$

On the other hand, we prove by induction on p that Algorithm 2.9 (De Casteljau) with input $\delta_{p,i}$ outputs the list $\delta'_{p,i}$ starting with i zeroes and with $(j+1)$-th element $\binom{j}{i}\alpha^{j-i}\beta^i$ for $j = i, \ldots, p$.

The result is clear for $p = i = 0$. If Algorithm 2.9 (De Casteljau) applied to $\delta_{p-1,i-1}$ outputs $\delta'_{p-1,i-1}$, the equality

$$\binom{j}{i}\alpha^{j-i}\beta^i = \alpha\binom{j-1}{i}\alpha^{j-i-1}\beta^i + \beta\binom{j-1}{i-1}\alpha^{j-i}\beta^{i-1}$$

proves by induction on j that Algorithm 2.9 (De Casteljau) applied to $\delta_{p,i}$ outputs $\delta'_{p,i}$. So the coefficients of $B_{p,i}(c,d)$ in the Bernstein's basis of e, d coincide with the output of Algorithm 2.9 (De Casteljau) with input $\delta_{p,i}$.

□

Notation 2.10 We denote by $\tilde{a}$ the list obtained by reversing the list a.

Algorithm 2.9 (De Casteljau) works both ways.

Corollary 2.11 *Let b b' and b'' be the lists of coefficients of P in the Bernstein's basis of c, d; e, d; and c, e respectively.*

Algorithm 2.9 (De Casteljau) applied to b with weights

$$\alpha = \frac{e-c}{d-c}, \beta = \frac{d-e}{d-c}$$

outputs b' and b''.

Algorithm 2.9 (De Casteljau) applied to b' with weights

$$\alpha' = \frac{c-e}{d-e}, \beta' = \frac{d-c}{d-e}$$

outputs b and $\tilde{b}''$.

Algorithm 2.9 (De Casteljau) applied to b'' with weights

$$\alpha'' = \frac{d-c}{e-c}, \beta'' = \frac{e-d}{e-c}$$

outputs $\tilde{b}'$ and b

Algorithm 2.9 (De Casteljau) gives a geometric construction of the control polygon of P on $[c, e]$ and on $[e, d]$ from the control polygon of P on $[c, d]$. The points of the new control polygons are constructed by taking iterated barycenters with weights α and β.

Example 2.12 Continuing Example 2.7, Algorithm 2.9 (De Casteljau) gives the following results.

4 -6 7 10
-1 1/2 17/2
-1/4 9/2
17/8

We construct the control line of P on $[0, 1/2]$ from the control line of P on $[0, 1]$ as explained in Figure 3 (see Annex).

We then draw the graph of P on $[0, 1]$ and the control line on $[0, 1/2]$ in Figure 4 (see Annex).

3 Real root isolation in the Bernstein basis

Let P be a polynomial of degree p in $\mathbb{R}[X]$. We are going to explain how to characterize the roots of P in $\mathbb{R}$, performing exact computations. The roots of P in $\mathbb{R}$ will be described by intervals with rational end points. Our method will be based on Descartes's law of signs (Theorem 1.1) and the properties of the Bernstein's basis studied in the preceding section.

Proposition 3.1 *Let b, b' and b'' be the lists of coefficients of P in the Bernstein's basis of c, d; e, d; and c, e. If $c < e < d$,then*

$$\mathrm{V}(b') + \mathrm{V}(b'') \leq \mathrm{V}(b).$$

Moreover if $P(e) \neq 0$, $\mathrm{V}(b) - \mathrm{V}(b') - \mathrm{V}(b'')$ is even.

Proof: The proof of the proposition is based on the following easy observations:

Inserting in a list $a = a_0, \ldots, a_n$ a value x in $[a_i, a_{i+1}]$ if $a_{i+1} \geq a_i$ (resp. in $[a_{i+1}, a_i]$ if $a_{i+1} < a_i$) between a_i and a_{i+1} does not modify the number of sign variations.

Removing from a list $a = a_0, \ldots, a_n$ with first non-zero $a_k, k \geq 0$, and last non-zero $a_\ell, k \leq \ell \leq n$, an element a_i, $i \neq k, i \neq \ell$ decreases the number of sign variation by an even (possibly zero) natural number.

Indeed the lists

$$\begin{aligned}
b = b_0^{(0)}, \ldots, \ldots, \ldots, \ldots, \ldots, b_p^{(0)} \\
b^{(1)} = b_0^{(0)}, b_0^{(1)}, \ldots, \ldots, \ldots, \ldots, b_{p-1}^{(1)}, b_p^{(0)} \\
\ldots \\
b^{(i)} = b_0^{(0)}, \ldots, \ldots, b_0^{(i)}, \ldots, \ldots, b_{p-i}^{(i)}, \ldots, \ldots, b_p^{(0)} \\
\ldots \\
b^{(p-1)} = b_0^{(0)}, \ldots, \ldots, \ldots, \ldots, b_0^{(p-1)}, b_1^{(p)}, \ldots, \ldots, \ldots, \ldots, b_p^{(0)} \\
b^{(p)} = b_0^{(0)}, \ldots, \ldots, \ldots, \ldots, \ldots, \quad b_0^{(p)} \quad , \ldots, \ldots, \ldots, \ldots, \ldots, b_p^{(0)}
\end{aligned}$$

are successively obtained by inserting intermediate values and removing elements that are not end points, since when $c < e < d$, $b_j^{(i)}$ is between $b_j^{(i-1)}$ and $b_{j+1}^{(i-1)}$, for $i = 1, \ldots, p, j = 0, \ldots, p-i-1$. Thus $\mathrm{V}(b^{(p)}) \leq \mathrm{V}(b)$ and the difference is even. Since

$$\begin{aligned}
b' = b_0^{(0)}, \ldots, \ldots, \ldots, \ldots, \ldots, \quad b_0^{(p)}, \\
b'' = b_0^{(p)} \quad , \ldots, \ldots, \ldots, \ldots, \ldots, b_p^{(0)},
\end{aligned}$$

$\mathrm{V}(b') + \mathrm{V}(b'') \leq \mathrm{V}(b^{(p)})$, and $\mathrm{V}(b') + \mathrm{V}(b'') \leq \mathrm{V}(b)$. If $P(e) \neq 0$, it is clear that $\mathrm{V}(b^{(p)}) = \mathrm{V}(b') + \mathrm{V}(b'')$, since $b_0^{(p)} = P(e) \neq 0$. □

Example 3.2 Continuing Example 2.12, we observe, denoting by b, b' and b'', the lists of coefficients of P in the Bernstein's basis of $0, 1$, $0, 1/2$, and $1/2, 1$, that $\mathrm{V}(b) = 2$. This is visible on Figure 1: the control line for $[0, 1]$ cuts twice the X-axis. Similarly, $\mathrm{V}(b') = 2$. This is visible on Figure 4: the control line for $[0, 1/2]$ also cuts twice the X-axis. Similarly, it is easy to check that $\mathrm{V}(b'') = 0$.

We cannot decide from this information whether P has two roots on $[0, 1/2]$ or no root on $[0, 1/2]$.

Let $P \in \mathbb{R}[X]$ and let b be the list of coefficients of P in the Bernstein's basis of c, d. We now describe a special case where the number $\mathrm{V}(b)$ coincides with the number of roots of P on (c, d). Let $d > c$, let $\mathcal{C}(c, d)_0$ be the closed disk with center $(c, 0)$ and radius $d - c$, and let $\mathcal{C}(c, d)_1$ be the closed disk with center $(d, 0)$ and radius $d - c$.

Suppose that $P \in \mathbb{R}[X]$ is a polynomial of degree p with all its real zeroes in $(-2^\ell, 2^\ell)$ and is squarefree. Consider natural numbers k and c such that $0 \leq c \leq 2^k$ and define

$$a_{c,k} = \frac{-2^{\ell+k} + c2^{\ell+1}}{2^k}.$$

It is clear that, for k big enough, the polynomial P has at most one root in $(a_{c,k}, a_{c+1,k})$ and has no other complex root in $\mathcal{C}(a_{c,k}, a_{c+1,k})_0 \cup \mathcal{C}(a_{c,k}, a_{c+1,k})_1$. Let $b(P, c, k)$ denote the list of coefficients of P in the Bernstein basis of $(a_{c,k}, a_{c+1,k})$. Note that $b(P, 0, 0)$, the list of coefficients of P in the Bernstein basis of $(-2^e ll, 2^\ell)$, can easily be computed from P, using Proposition 2.3.

Using Theorem 2.5, it is possible to decide, for k big enough, whether P has exactly one root in $(a_{c,k}, a_{c+1,k})$ or has no root on $(a_{c,k}, a_{c+1,k})$ by testing whether $\mathrm{V}(b(P, c, k))$ is zero or one.

Example 3.3 Continuing Example 3.2, let us study the roots of P on $(0, 1)$, as a preparation to a more formal description of Algorithm Algorithm 3.4 (Real Root Isolation).

The Bernstein's coefficients of P for $(O, 1)$ are $4, -6, 7, 10$. There maybe roots of P on $(0, 1)$ as there are sign variations in its Bernstein's coefficients.

As already seen in Example 3.2,a first application of Algorithm 2.9 (De Casteljau) with weights $1/2, 1/2$ gives

4		-6		7		10	
	-1		1/2		17/2		
		-1/4		9/2			
			17/8				

There maybe roots of P on $(0, 1/2)$ as there are as there are sign variations in the Bernstein's coefficients of P which are $32, -8, -2, 17$. There are no roots of P on $(1/2, 1)$.

Let us apply once more Algorithm 2.9 (De Casteljau) with weights $1/2, 1/2$:

4		-1		-1/4		17/8	
	3/2		-5/8		15/16		
		7/16		5/32			
			19/64				

There are no sign variations on the sides of the triangle so there are no roots of P on $(0, 1/4)$ and on $(1/4, 1/2)$.

An **isolating list for** P is a finite list L of rational points and disjoint open intervals with rational end points of $\mathbb{R}$ such that each point or interval of L contains exactly one root of P in $\mathbb{R}$ and every root of P in $\mathbb{R}$ belongs to an element of L.

Algorithm 3.4 (Real Root Isolation)

Input: *the list $b(P, 0, 0)$ the Bernstein coefficients of a squarefree non zero polynomial $P \in \mathbb{R}[X]$ for $(-2^\ell, 2^\ell)$, where $(-2^\ell, 2^\ell)$ is an interval containing the roots of P in $\mathbb{R}$.*

Output: *a list $L(P)$ isolating for P.*

Procedure:

Initialization: Define $Pos := \{b(P, 0, 0)\}$ *and* $L(P) := \emptyset$.

While Pos *is non-empty,*

Remove $b(P, c, k)$ *from* Pos.

If $\mathrm{V}(b(P, c, k)) = 1$, *add* $(a_{c,k}, a_{c+1,k})$ *to* $L(P)$.

If $\mathrm{V}(b(P, c, k)) > 1$,

Compute $b(P, 2c, k + 1)$ *and* $b(P, 2c + 1, k + 1)$ *using Algorithm 2.9 (De Casteljau) with weights* $(1/2, 1/2)$ *and add them to* Pos.

If $P(a_{2c+1,k+1}) = 0$, *add* $\{a_{2c+1,k+1}\}$ *to* $L(P)$.

Output $L(P)$.

The hypotheses are not a real loss of generality since, given any polynomial Q, a squarefree polynomial P having the same roots at Q can be computed using the gcd of Q and Q' (see for example [1]).

Moreover denoting

$$Q = c_p X^p + \ldots + c_0$$
$$C(Q) = \sum_{0 \le i \le p} \left| \frac{c_i}{c_p} \right|,$$

the absolute value of any root of Q in $\mathbb{R}$ is smaller than $C(Q)$ [5, 1], so that it is easy, knowing Q, to compute ℓ such that $(-2^\ell, 2^\ell)$ contains the roots of Q in $\mathbb{R}$.

Using classical bounds on the minimal distance between the roots of a polynomial Q with integer coefficients [5, 1] one can prove that the binary complexity of computing the squarefree part P of Q, computing ℓ such that $(-2^\ell, 2^\ell)$ contains the roots of Q in $\mathbb{R}$, and performing Algorithm 3.4 (Real Root Isolation) for P, is $O(p^6(\tau + log_2(p))^2)$, where p is a bound on the degree of Q and τ a bound on the bit size of the coefficients of Q [1]. The coefficients of the elements of the $b(P, c, k)$ computed in the algorithm are rational numbers of bit size $O(p^2(\tau + \log_2(p)))$ [1]. Since there are at most $2p$ values of $b(P, c, k)$ in Pos throughout the computation, and there are $p + 1$ coefficients in each $b(P, c, k)$,the workspace of the algorithm is $O(p^4(\tau + \log_2(p)))$.

Experiments shows that the algorithm is very efficient and, in may cases, the computations do not end because of the memory consuming.

An improved (in terms of memory) version of Algorithm 3.4 (Real Root Isolation) is based on the following idea, inspired from [8]: since every $b(P, c, k)$ computed in the algorithm carries the whole information about P, it is not necessary to store the value of $b(P, c, k)$ at all the nodes, and the workspace of the algorithm can be improved.

It will be necessary to convert the Bernstein's coefficients of P on an interval $(a_{d,m}, a_{d+1,m})$ into the Bernstein's coefficients of P on an interval $(a_{c,k}, a_{c+1,k})$ in the special case $k > m, a_{c,k} \ge a_{d,m}$.

Algorithm 3.5 (Convert)

Input: (c,k), (d,m) *with* $k \geq m, a_{c,k} \geq a_{d,m}$ *and the Bernstein coefficients* $b(P,d,m)$ *of* P *on* $(a_{d,m}, a_{d+1,m})$.

Output: *the Bernstein coefficients* $b(P,c,k)$ *of* P *on* $(a_{c,k}, a_{c+1,k})$.

Procedure:

Define e, n *with* $e = e_0, \ldots, e_{n-1}$ *such that*

$$c = e_0 \ldots e_{n-1} c_n \ldots c_{k-1},$$
$$d = e_0 \ldots e_{n-1} d_n \ldots d_{m-1},$$

are the binary representations of c *and* d *with* $c_n \neq d_n$ *and* $b := b(P,d,m)$.

For every i *from* n *to* $m-1$

If $d_i = 0$ *apply Algorithm 2.9 (De Casteljau) with weights* $(-1,2)$ *and output* b', b''. *Update* $b := b'$.

If $d_i = 1$ *apply Algorithm 2.9 (De Casteljau) with weights* $(2,-1)$ *and output* b', b''. *Update* $b := b''$.

For every i *from* n *to* $k-1$

If $c_i = 0$ *apply Algorithm 2.9 (De Casteljau) with weights* $(1/2, 1/2)$ *and output* b', b''. *Update* $b := b'$.

If $c_i = 1$ *apply Algorithm 2.9 (De Casteljau) with weights* $(1/2, 1/2)$ *and output* b', b''. *Update* $b := b''$.

Output b.

The correctness of Algorithm 3.5 (Convert) follows clearly from the correctness of Algorithm 2.9 (De Casteljau).

It is now easy to described the improved real root isolation method.

Algorithm 3.6 (Improved Real Root Isolation)

Input: *the list* $b(P,0,0)$ *the Bernstein coefficients of a squarefree non zero polynomial* $P \in \mathbb{R}[X]$ *for* $(-2^\ell, 2^\ell)$, *where* $(-2^\ell, 2^\ell)$ *is an interval containing the roots of* P *in* $\mathbb{R}$.

Output: *a list* $L(P)$ *isolating for* P.

Procedure:

Initialization: Define $Pos := [(0,0)]$, $L(P) := \emptyset$, $d := 0$,$m := 0$.

While Pos *is non-empty,*

Remove the first element (c,k) *of* Pos.

Compute $b(P,c,k)$ *from* $b(P,d,m)$ *using Algorithm 3.5 (Convert).*

If $\mathrm{V}(b(P,c,k)) = 1$, *add* $(a_{c,k}, a_{c+1,k})$ *to* $L(P)$.

If $\mathrm{V}(b(P,c,k)) > 1$,

$Pos = [(2c, k+1), (2c+1, k+1)] \bigcup Pos$

If $P(a_{2c+1,k+1}) = 0$, $L(P) = L(P) \bigcup \{a_{2c+1,k+1}\}$

Update $d := c$, $m := k$.

Output $L(P)$.

One can easily see that the nodes (c,k) are computed with respect to the lexicographic ordering (denoted by $<$ in this paragraph). Also, when (c,k) and (c',k') are two consecutive nodes, with $(c,k) < (c',k')$, then Algorithm 2.9 est applied only once by Algorithm 3.5. The only extra cost will occur in the case where $(c,k) > (c',k')$. According to [4], the number of such jumps is low compared with the total number of nodes.

The binary complexity of computing the squarefree part P of Q, computing ℓ such that $(-2^\ell, 2^\ell)$ contains the roots of Q in $\mathbb{R}$, and performing Algorithm 3.6 (Improved Real Root Isolation) for P, is $O(p^6(\tau + log_2(p))^2)$, where p is a bound on the degree of Q and τ a bound on the bit size of the coefficients of Q, since every node in the tree created by Algorithm 3.4 (Real Root Isolation) is visited at most three times in Algorithm 3.6 (Improved Real Root Isolation). However Algorithm 3.6 (Improved Real Root Isolation) uses only an $O(p^3(\tau + \log_2(p)))$ workspace, since only one vector of Bernstein's coefficients is stored throughout the computation in Algorithm 3.6 (Improved Real Root Isolation), rather than possible $2v$ in the Algorithm 3.4 (Real Root Isolation).

We can also perform the computation using interval arithmetic. The advantages of interval arithmetic is that it is much quicker than exact arithmetic when the end-points of the intervals are represented, for example, by floating point numbers with a fixed precision, and it allows computations with polynomials known approximately.

The basic idea of interval arithmetic is that real numbers are represented by intervals with rational bounds encoded as floating point numbers with a fixed precision and a convenient arithmetic is defined on these intervals. Fixing a precision u, n, the u, n-intervals are of the form $[\frac{i}{2^{n_i}}, \frac{j}{2^{n_j}}]$, with i and j being integers between -2^u and 2^u, $i \leq j$, and n_i and n_j being integers between -2^n and 2^n. A consistent interval arithmetic will be compatible with the arithmetics operations : if α and β are two real number represented respectively by two intervals $[r_\alpha, l_\alpha]$ and $[r_\beta, l_\beta]$, the result of any operation $[r_\alpha, l_\alpha]$ op $[r_\beta, l_\beta]$, will contain the real number α op β. In the next paragraph, we assume working with a multi-precision interval arithmetics such as in [7] (where u can be arbitrary fixed by the user).

Thus the correctness of the interval arithmetic operations when performing Algorithm 3.5 (Convert) comes from the fact that if the Bernstein coefficients of P for (c,d) are bounded by M, the Bernstein coefficients of P for (e,f), $c \leq e < f \leq d$ are also bounded by M.

The sign of an interval $[a,b], a \leq b$ is defined as follows:

$$\text{sign}([a,b]) = \begin{cases} 0 & \text{if } a = b = 0, \\ 1 & \text{if } a > 0, \\ -1 & \text{if } b < 0, \\ ? & \text{if } a \leq 0 \leq b,\ a \neq 0,\ b \neq 0. \end{cases}$$

If sign$([a,b]) \neq ?$, we say that the sign of $[a,b]$ is well defined. The numbers of sign variations in a list $[a_0,b_0],\cdots,[a_p,b_p]$ of intervals with rational end points is defined as follows:

- If for all $i=0,\cdots,p$, sign$([a_i,b_i]) \neq ?$,
$$\mathrm{V}([a_0,b_0],\cdots,[a_p,b_p]) = \mathrm{V}(\text{sign}([a_0,b_0]),\cdots,\text{sign}([a_p,b_p])).$$
- If for every $i=1,\cdots,p$, such that sign$([a_i,b_i]) =?$, sign$([a_{i-1},b_{i-1}])$ and sign$([a_{i+1},b_{i+1}])$ are well defined,
$$\mathrm{V}(a) = \mathrm{V}(b)$$
where b is obtained by removing from a all the $[a_i,b_i]$ such that sign$([a_i,b_i]) =?$.
- Otherwise $\mathrm{V}(a) =?$.

Example 3.7 *If* $a=[1,2],[-1,1]$, $\mathrm{V}(a)=?$
If $b=[1,2],[-1,1],[-2,-1]$, $\mathrm{V}(b)=1$.

Algorithm 3.8 (Interval-arithmetic Real Root Isolation)

Input: *a precision u,n, a list $\bar{b}(0,0)$ of $p+1$ u,n-intervals whose first and last element do not contain 0.*

Output: *a list L and a list N of intervals such that for every ℓ and every polynomial P whose Bernstein coefficients for $(-2^\ell,2^\ell)$ belong to $\bar{b}(0,0)$, there exists one and only one root of P in each interval of L and all the other roots of P in $(-2^\ell,2^\ell)$, belong to an interval of N.*

Procedure:

Initialization: Compute $\mathrm{V}(\bar{b}(0,0))$, using u,n-arithmetic, define $Pos := \{(0,0)\}$, $L(P) := \emptyset$, $N(P) := \emptyset$, $d := 0$,$m := 0$.

While Pos is non-empty,

Remove the first element (c,k) of Pos.
Compute $\bar{b}(c,k)$ from $\bar{b}(d,m)$ by Algorithm 3.5 (Convert), using u,n-arithmetic.
If $\mathrm{V}(\bar{b}(c,k))=1$, add $(a_{c,k},a_{c+1,k})$ to L.
If $\mathrm{V}(\bar{b}(c,k))>1$, Add $(2c,k+1),(2c+1,k+1)$ at the beginning of Pos.
If $\mathrm{V}(\bar{b}(c,k)) =?$, add $(a_{c,k},a_{c+1,k})$ to N.
Update $d := c$, $m := k$.

Output L,N.

Note that the interval arithmetic can be used as well when the polynomial P is known exactly. In this case we can compute the square free part of P and it is easy to design a variant of Algorithm 3.8 and output a list of isolating intervals by augmenting precision, examining again the intervals where no decision has been taken yet.

Algorithm 3.9 (Interval-arithmetic Exact Real Root Isolation)

Input: *a square free $P \in \mathbb{R}[X]$ and the list $b(P,0,0)$ of Bernstein coefficients of P for $(-2^\ell, 2^\ell)$, where $(-2^\ell, 2^\ell)$ contains the roots of P in $\mathbb{R}$.*

Output: *a list $L(P)$ isolating for P.*

Procedure:

Initialization: u such that the elements of $b(P,0,0)$ belong to $(-2^u, 2^u)$, $n := 1$. Compute $\mathrm{V}(b(P,0,0))$, define $Pos := \{(0,0)\}$, $L(P) := \emptyset$, $N(P) := \emptyset$, $d := 0$,$m := 0$.

$(\star)$ *While Pos is non-empty,*

Remove the first element (c,k) of Pos.

Compute $b(P,c,k)$ from $b(P,d,m)$ by Algorithm 3.5 (Convert) using u, n-arithmetic.

If $\mathrm{V}(b(P,c,k)) = 1$, add $(a_{c,k}, a_{c+1,k})$ to $L(P)$.

If $\mathrm{V}(b(P,c,k)) > 1$,

Add $(2c, k+1), (2c+1, k+1)$ at the beginning of Pos.

If $P(a_{2c+1,k+1}) = 0$, add $\{a_{2c+1,k+1}\}$ to $L(P)$.

If $\mathrm{V}(b(P,c,k)) =?$, add $(a_{c,k}, a_{c+1,k})$ to $N(P)$.

Update $d := c$, $m := k$.

If $N(P) \neq$, update $n := n+1$, $Pos = N(P)$, go to $(\star)$.

Output $L(P)$.

The experimental complexity of Algorithms 3.8 and 3.9 is good, and real root isolation can be performed by this method for polynomials of degree several thousands and with coefficients of bit size several hundreds.

4 Real root isolation in the monomial basis

The preceding method for real root isolation is adapted to polynomials expressed in the Bernstein basis. We describe now variants adapted to the case when the polynomial is expressed in the monomial basis. Such algorithms have been already studied extensively, starting from [9]. Our presentation is based on [8].

Rather than looking at the Bernstein coefficients of the same polynomial P on varying intervals, we are going to consider different polynomials closely related to P on each interval. We need some notation. Suppose as before that $P \in \mathbb{R}[X]$ is a polynomial of degree p with all its real zeroes in $(-2^\ell, 2^\ell)$ and is squarefree, consider natural numbers k and c such that $0 \leq c \leq 2^k$ and define

$$a_{c,k} = \frac{-2^{\ell+k} + c2^{\ell+1}}{2^k}.$$

We define

$$P_{c,k} := \mathrm{C}_{2^{\ell+1-k}}(\mathrm{T}_{-a_{c,k}}(P)).$$

The polynomial $P_{c,k}$ is simply the result of the transformation operated on P when the segment $(a_{c,k}, a_{c+1,k})$ is sent to the segment $(0,1)$ by a translation followed by a contraction.

All the algorithms of the preceding section have analogous versions in the Bernstein's basis. We describe only the algorithms corresponding to the conversion from one interval to another and the improved root isolation algorithm, it is also possible to use interval arithmetic in a similar manner in the monomial basis.

Algorithm 4.1 (Change interval)

Input: (c,k), (d,m) *with* $k \geq m, a_{c,k} \geq a_{d,m}$ *and the polynomial* $P_{d,m}$.

Output: *the polynomial* $P_{c,k}$

Procedure:

Define e, n *with* $e = e_0, \ldots, e_{n-1}$ *such that*

$$c = e_0 \ldots e_{n-1} c_n \ldots c_{k-1},$$
$$d = e_0 \ldots e_{n-1} d_n \ldots d_{m-1},$$

are the binary representations of c *and* d *with* $c_n \neq d_n$, *and* $R := P_{d,m}$.

For every i *from* n *to* $m-1$

If $d_i = 0$, $R := \mathrm{C}_2(R)$.
If $d_i = 1$, $R := \mathrm{C}_2(\mathrm{T}_{-1}(R))$.

For every i *from* n *to* $k-1$

If $c_i = 0$, $R := \mathrm{C}_{1/2}(R)$
If $c_i = 1$, $R := \mathrm{C}_{1/2}(\mathrm{T}_{-1/2}(R))$

Output R.

The correctness of the algorithm follows clearly from the definition of $P_{c,k}$.
It is now easy to describe the improved real root isolation method in the monomial basis.

Algorithm 4.2 (Real Root Isolation in monomial basis)

Input: *a squarefree non zero polynomial* $P \in \mathbb{R}[X]$, *and* $(-2^\ell, 2^\ell)$ *an interval containing the roots of* P *in* $\mathbb{R}$.

Output: *a list* $L(P)$ *isolating for* P.

Procedure:

Initialization: Define $Pos := \{(0,0)\}$, $L(P) := \emptyset$, $d := 0$,$m := 0$.

While Pos *is non-empty,*

Remove the first element (c,k) *of* Pos.

Compute $P_{c,k}$ *from* $P_{d,m}$ *using Algorithm 3.5 (Change Interval). Take* $Q_{c,k} := \mathrm{T}_{-1}\left(\mathrm{Rec}_p(P_{c,k})\right)$.

If $\mathrm{V}(Q_{c,k}) = 1$, *add* $(a_{c,k}, a_{c+1,k})$ *to* $L(P)$.

If $\mathrm{V}(Q_{c,k}) > 1$,

Add $(2c, k+1), (2c+1, k+1)$ *at the beginning of* Pos.

If $P(a_{2c+1,k+1}) = 0$, *add* $\{a_{2c+1,k+1}\}$ *to* $L(P)$.

Update $d := c$, $m := k$.

Output $L(P)$.

The correctness of Algorithm 4.2 follows from the correctness of Algorithm 3.6, noting that $\mathrm{V}(Q_{c,k}) = \mathrm{V}(b(P,c,k))$ by Proposition 2.3. It has been shown in [8] that in any case Algorithm 4.2 performs at most one translation of kind $T_{-1/2}$ or T_{-1}.

The complexity analysis and experimental behavior of the real root isolation method in the monomial basis is quite similar to the one in the Bernstein basis.

5 Conclusion

We have described to versions of an algorithm based on Descarte's rule of sign for isolating the real roots of a univariate polynomial. They differ by the representation used for the polynomials : Bernstein basis or monomial basis. The efficiency of these methods have already been shown in [8] and [6]. One major problem of such strategies is the memory consuming in regards to the efficiency in terms of bit operations. The "optimal" solution proposed in [8] for the case of polynomials represented in the monomial basis has been adapted here to the case of polynomials represented in the Bernstein Basis.

References

[1] S. BASU, R. POLLACK,M.-F. ROY, *Algorithms in real algebraic geometry*, Springer (2003).

[2] R. DESCARTES, *Géométrie* (1636). A source book in Mathematics, 90-31. Harvard University press (1969).

[3] G. FARIN, *Curves and surfaces for Computer Aided Design*, Academic Press (1990).

[4] KRANDICK, W. Trees and Jumps and Real Roots Journal of Computational and Applied Mathematics, 2003.

[5] M. MIGNOTTE, D. STEFANESCU *Polynomials, an algorithmic approach*, Springer Verlag, Singapore (1999).

[6] B. MOURRAIN, M. N. VRAHATIS, J.-C. YAKHOUBSON *On the Complexity of Isolating Real Roots and Computing with Certainty the Topological Degree*, Journal of Complexity, 182, 612–640 (2002).

[7] N. REVOL AND F. ROUILLIER, *Motivations for an Arbitrary Precision Interval Arithmetic and the MPFI Library*, Workshop on Validated Computing, Toronto - Canada, 155-161 (2002).

[8] F. ROUILLIER, P. ZIMMERMANN, *Efficient Isolation of Polynomial Real Roots*, Journal of Computational and Applied Mathematics, 162 (1), 33-50 (2003).

[9] J.V. USPENSKY, *Theory of equations*, MacGraw Hill (1948).

Annex

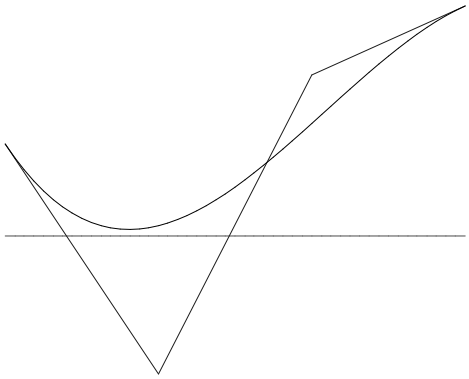

Figure 1: Graph of P on $[0, 1]$.

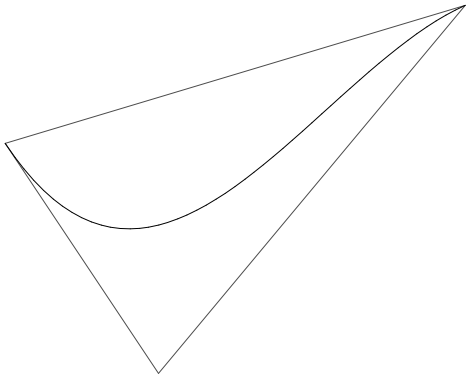

Figure 2: Graph of P on $[0, 1]$ and the control polygon.

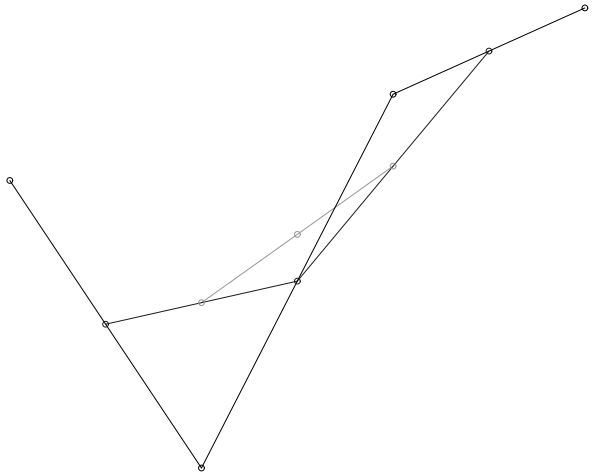

Figure 3: Control line of P on $[0, 1/2]$.

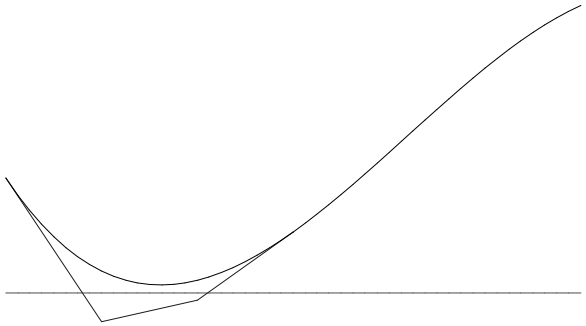

Figure 4: Graph of P on $[0,1]$ and control line on $[0,1/2]$.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399