Yugoslav Journal of Operations Research
24 (2014) Number 1,53-69
DOI:10.2298/YJOR120620004L

COMPUTING REAL ZEROS OF A POLYNOMIAL BY BRANCH AND BOUND AND BRANCH AND REDUCE ALGORITHMS

Hoai An LE THI
Laboratoire d'Informatique Théorique et Appliquée (LITA)
Université de Lorraine, 57045 Metz cedex1-France
hoai-an.le-thi@univ-lorraine.fr

Mohand OUANES
Département de Mathématiques, Faculté des Sciences,
Université de Tizi-Ouzou, Algérie
mohand.Ouanes@yahoo.fr

Ahmed ZIDNA
Laboratoire d'Informatique Théorique et Appliquée (LITA)
Université de Lorraine, 57045 Metz cedex1-France
ahmed.zidna@univ-lorraine.fr

Received: June 2012 / Accepted: December 2013

Abstract: In this paper we propose two algorithms based on branch and bound method and reduced interval techniques to compute all real zeros of a polynomial. Quadratic bounding functions are proposed which are better than the well known linear underestimator. Experimental result shows the efficiency of the two algorithms when facing ill-conditionned polynomials.

Keywords: Global optimization quadratic upper function quadratic lower function root-finding Bound and Reduce Branch and Bound w-subdivision
MSC:26C10, 90C20, 90C25, 90C90.

1 INTRODUCTION

Several fundamental geometrical problems that arise in the processing of curves and surfaces may be reduced computationally by isolating and approximating the distinct real roots of univariate polynomials on finite intervals. Many different approaches for solving a polynomial equation exist [1]. We briefly mention the methods based on deflation techniques [2]. Other ones proceed by subdividing the interval into a sequence of intervals such that each one contains one and only one zero of the polynomial [3]. In [7], the authors propose a method for finding real zeros of a polynomial in Bernstein basis. In recent years univariate global optimization problems have attracted common attention because they arise in many real-life applications and the obtained results can be easily generalized to multivariate case. Let us mention the works for the Polynomial and Rational functions [13], [21], the Lipschitz functions [15], and those in [9], [12], [20], [22]. Root-finding problem is not an optimization problem, however we can exploit the idea of branch and bound techniques in global optimization for finding zeros of a polynomial.
In this paper we propose two approaches for finding all real zeros of a polynomial in a power basis:

1. A Bound and Reduce approach (BR):

   The main idea consists in constructing quadratic underestimation and/or overestimation functions of the given polynomial f in a successive reduced interval $[a_k, b_k]$, in the way that the zeros of the quadratic function and the zeros of the polynomial f are the same.

2. An adapted Branch and Bound method (BB):

   The main idea is to localize the intervals containing the zeros of the polynomial f by constructing quadratic underestimation and/or overestimation functions [9, 10]. In fact, the minimum or the maximum respectively of the lower or the upper bound function is used to subdivide the initial interval into two sub-intervals on which the polynomial is less variying. The process is stoped when the current interval has at most one zero.

Let Lf_k and Uf_k be a lower and a upper bound of f on $[a_k, b_k]$. The common procedures of both approaches are:

- if $Lf_k(x) > 0$, then $f(x) > 0 \quad \forall x \in [a_k, b_k]$. This means that the polynomial has no zero in this interval;

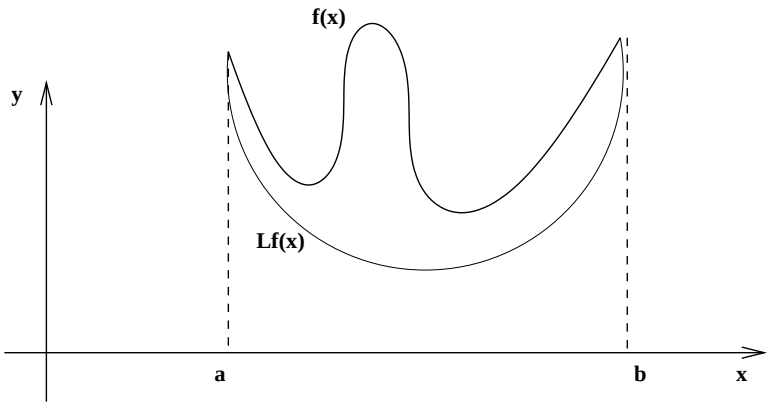

Figure 1: The lower bound Lf(x) is strictly positive on $[a, b]$, the polynomial $f(x)$ has no zero in this interval.

- if $Uf_k(x) < 0$, then $f(x) < 0 \quad \forall x \in [a_k, b_k]$ and so the polynomial has no zeros in this interval;

- if $Lf_k(x)$ or $Uf_k(x)$ has one or two roots on the current interval $[a_k, b_k]$ which are not the zeros of the polynomial, then these roots are used as ends to reduce the current interval. By the way, when reducing the interval containing all the zeros of f , we can locate all sub-intervals that contain the zeros of f. These zeros are in fact the zeros of quadratic underestimating and/or overestimating functions of f on these sub-intervals.

The performance of the proposed procedure depends on the quality of the chosen lower and upper bounds of f. We introduce a quadratic lower bounding function which is better than the well known linear underestimating of f by the theory of approximation [6]. In the same way we introduce a quadratic upper bounding function of f.
The structure of the rest of the paper is as follows: Section 2 discusses the construction of a lower and an upper bound of a polynomial. Section 3 describes an Bound and Reduce (BR) algorithm to approximate the real zeros of a polynomial. Section 4 describes a branch and bound algorithm (BB) and Section 5 presents some numerical examples for ill-conditionned polynomials while Section 6 contains some conclusions.

2 QUADRATIC BOUNDING FUNCTIONS

We now explain how to construct an upper bound of a function f which is twice continuously differentiable on an interval $[a, b]$. We assume that there exists a positive number K such that $|f''(x)| \leq K$ for all $x \in [a, b]$.

For $m \geq 2$, let $\{w_1, w_2, ..., w_m\}$ be the pairwise functions defined as in [6]:

$$w_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & \text{if } x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

We have

$$\sum_{i=1}^{i=m} w_i(x) = 1, \forall x \in [a, b] \text{ and } w_i(x_j) = 0 \text{ if } i \neq j, \ \ 1, \text{ otherwise.}$$

Let $L_h f$ be the piecewise linear interpolant to f at the points $x_1, x_2, ..., x_m$:

$$L_h f(x) = \sum_{i=1}^{i=m} f(x_i) w_i(x). \tag{1}$$

The next result from [6] gives an upper and a lower bound of f on the interval $[a, b]$, $(h = b - a)$.

[6] For all $x \in [a, b]$, we have $|L_h f(x) - f(x)| \leq \frac{1}{8} K h^2$, i.e.,

$$L_h f(x) - \frac{1}{8} K h^2 \leq f(x) \leq L_h f(x) + \frac{1}{8} K h^2.$$

In [9] the following quadratic lower bounding function of f is proposed:

$$Lf(x) := L_h f(x) - \frac{1}{2} K (x - a)(b - x) \leq f(x), \ \forall x \in [a, b] .$$

It has been proved (see [9]) that this lower bound is better than the affine minorization given in [6]:

$$Lf(x) \geq L_h f(x) - \frac{1}{8} K h^2.$$

In a similar way, we now introduce a concave quadratic upper bounding function of f :

For all $x \in [a, b]$, we have

$$L_h f(x) + \frac{1}{8} K h^2 \geq Uf(x) := L_h f(x) + \frac{1}{2} K (x - a)(b - x) \geq f(x). \tag{2}$$

Proof. Let $E(x)$ be the function defined on $[a, b]$ by

$$\begin{aligned} E(x) &= L_h f(x) + \frac{1}{8}Kh^2 - Uf(x) && (3)\\ &= \frac{1}{8}Kh^2 - \frac{1}{2}K(x-a)(b-x) && (4)\\ &= \frac{K}{2}\left[x^2 - (a+b)x + ab + \frac{1}{4}(b-a)^2\right]. && (5) \end{aligned}$$

E is convex on $[a, b]$, and its derivative is equal to zero at $x^* = \frac{1}{2}(a+b)$. Therefore, for any $x \in [a, b]$ we have

$$E(x) \geq \min\{E(x) : x \in [a, b]\} = E(x^*) = 0. \quad (6)$$

Then, the first inequality in (2) holds. Consider now the function ϕ defined on $[a, b]$ by

$$\phi(x) := Uf(x) - f(x) = L_h(x) + \frac{1}{2}K(x-a)(b-x) - f(x). \quad (7)$$

It is clear that $\phi''(x) = -K - f''(x) \leq 0$ for all $x \in [a, b]$. Hence ϕ is a concave function, and for all $x \in [a, b]$ we have

$$\phi(x) \geq \min\{\phi(x) : x \in [a, b]\} = \phi(a) = \phi(b) = 0. \quad (8)$$

The second inequality in (2) is then proved.

3 BOUND AND REDUCE METHOD (BR)

In this section we describe a Bound and Reduce algorithm for approximating the real zeros of a polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$ in an interval $[a, b]$. The initial interval which contains all the zeros of f can be computed by using the Cauchy or the Knuth method. Let K be a positive number such that $|f''(x)| \leq K$, $\forall x \in [a, b]$. As described above, we construct upper bounds and lower bounds of f on successive reduced intervals $[a_k, b_k]$ of $[a, b]$. More precisely,

- If $f(a_k) > 0$, we construct Lf_k, a convex quadratic underestimating function of f on the interval $[a_k, b_k]$ defined by setting

$$Lf_k(x) = f(a_k)\frac{b_k - x}{h_k} + f(b_k)\frac{x - a_k}{h_k} - \frac{1}{2}K(x-a_k)(b_k-x). \quad (9)$$

  Clearly, if $Lf_k(x)$ has no roots in $[a_k, b_k]$, then $Lf_k(x) > 0$ $\forall x \in [a_k, b_k]$. Consequently, $f(x) > 0$ $\forall x \in [a_k, b_k]$. Hence $f(x)$ has no roots in $[a_k, b_k]$.

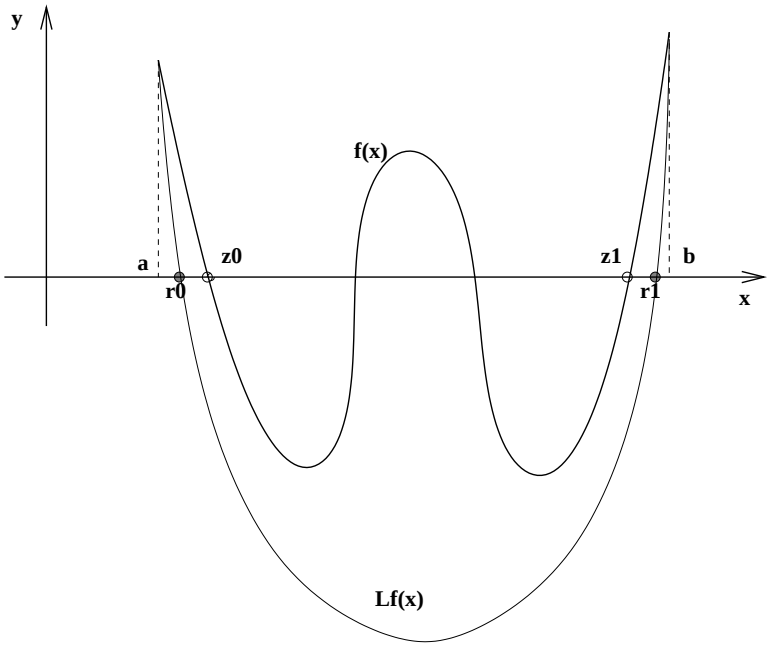

Figure 2: Root finding by Bound and Reduce the interval. The roots of Lower Bound function $Lf(x), r_0$ and r_1 approximate respectively the extremal root z_0 and z_1 of the polynomial $f(x)$.

- If $f(a_k) < 0$, we construct Uf_k, a concave quadratic overestimating function of f on the interval $[a_k, b_k]$, by setting

$$Uf_k(x) = f(a_k)\frac{b_k - x}{h_k} + f(b_k)\frac{x - a_k}{h_k} + \frac{1}{2}K(x - a_k)(b_k - x). \qquad (10)$$

  Similarly as in the above, if $Uf_k(x)$ has no roots in $[a_k, b_k]$, then $Uf_k(x) < 0$ $\forall x \in [a_k, b_k]$. So $f(x)$ has no roots in $[a_k, b_k]$.

The recursive algorithm can be given as follows :

3.1 Convergence of the algorithm

The algorithm terminates if one of the following criteria is satisfied:

1. The length of the current interval $[a_k, b_k]$ is less than ϵ ;

2. The lower or the upper bound of the polynomial has no zeros on the current interval $[a_k, b_k]$.

Algorithm 1: Branch and reduced algorithm

Function $S = \mathbf{ZeroPolynom}(f, n, a, b, \epsilon)$;

Data: f : the polynomial, n : the degree of the polynomial, a, b : the end of the interval $[a, b]$, ϵ : precision of the zeros

Result: S - the set of all found zeros of f

begin

$S_k = \emptyset$ is an intermediate set ;

if $(b - a) < \epsilon$ **then** $S = \emptyset$ **return** S;

;

Compute $f(a)$;

if $f(a) > 0$ **then**

Construct Lf_k, a quadratic lower bound of f on the interval $[a, b]$;

Solve the equation $Lf_k(x) = 0$;

if $Lf_k(x)$ *has no root in* $[a, b]$ **then**

$S_k = \emptyset$;

end

else if Lf_k *has one root* $r_1 \in [a, b]$, **then**

if $|f(r_1)| < \epsilon$, **then** $S_k = S_k \cup \{r_1\}$;

;

$S_k = S_k \cup ZeroPolynom(f, n, r_1 + \epsilon, b, \epsilon)$;

end

else if Lf_k *has two roots* $r_1 \in [a, b]$ *and* $r_2 \in [a, b]$, **then**

if $|f(r_1)| < \epsilon$ **then** $S_k = S_k \cup \{r_1\}$;

if $|f(r_2)| < \epsilon$ **then** $S_k = S_k \cup \{r_2\}$;

$S_k = S_k \cup RootPolynom(f, n, r_1 + \epsilon, r_2 - \epsilon, \epsilon)$

end

end

else

Construct Uf_k a quadratic upper bound of f on the interval $[a, b]$

Solve the equation $Uf_k(x) = 0$

if Uf_k *has no root in* $[a, b]$ **then** $S_k = \emptyset$;

;

else if Uf_k *has one root* $r_1 \in [a, b]$ **then**

if $|f(r_1)| < \epsilon$ **then** $S_k = S_k \cup \{r_1\}$;

;

$S_k = S_k \cup ZeroPolynom(f, n, r_1 + \epsilon, b, \epsilon)$**;**

end

else if Uf_k *has two roots* $r_1 \in [a, b]$ *and* $r_2 \in [a, b]$ **then**

if $|f(r_1)| < \epsilon$ **then** $S_k = S_k \cup \{r_1\}$;

if $|f(r_2)| < \epsilon$ **then** $S_k = S_k \cup \{r_2\}$;

$S_k = S_k \cup ZeroPolynom(f, n, r_1 + \epsilon, r_2 - \epsilon, \epsilon)$

end

end

$S = S_k$, **return** S;

end

For $\epsilon > 0$, at least one of the two above conditions must be satisfied after a finite number of iterations: if the second condition is violated during the algorithm, then the first condition must be fulfilled after at most $m = \lfloor (b-a)\sqrt{\frac{K}{8\varepsilon}} \rfloor + 1$ iterations (see [9]).

For $\epsilon = 0$, we have the following result.

For $h_k = b_k - a_k$, we have

$$\lim_{h_k \to 0} (Uf_k(x) - f(x)) = 0 \quad \text{and} \quad \lim_{h_k \to 0} (f(x) - Lf_k(x)) = 0.$$

Proof. As

$$0 \leq Uf_k(x) - f(x) \leq \frac{1}{2}K(s-a_k)(b_k-s) \leq \frac{1}{2}K{h_k}^2,$$

it holds

$$\lim_{h_k \to 0} (Uf_k(x) - f(x)) = 0.$$

In the same way, if we have

$$0 \leq f(x) - Lf_k(x) \leq \frac{1}{2}K(s-a_k)(b_k-s) \leq \frac{1}{2}K{h_k}^2,$$

then

$$\lim_{h_k \to 0} (f(x) - Lf_k(x)) = 0.$$

The proof is complete.

4 BRANCH AND BOUND METHOD (BB)

In this section we describe the Branch and Bound algorithm for approximating the real zeros of a polynomial in an interval $[a, b]$. The initial interval which contains all the zeros of f can be computed by using the Cauchy or the Knuth method. Let K be a positive number such that $|f''(x)| \leq K$, $\forall x \in [a, b]$.

As described above, we construct upper bounds and lower bounds of f on successive sub-intervals $[a_k, b_k]$ of $[a, b]$.

The main idea is to subdivide the initial interval into sub-intervals which contains at most one zero. At iteration k, for dividing $T^k = [a_k, b_k]$, one cane use its middle point (the normal subdivision). Due to the efficiency of the w-subdivision introduced in [9] we use this procedure in our BB algorithm (see Figure 3).

Using this idea, we divide $[a_k, b_k]$ via x_k^L the minimum of the lower bound or x_k^U the upper bound. This procedure seems to be efficient: we often obtain the exact

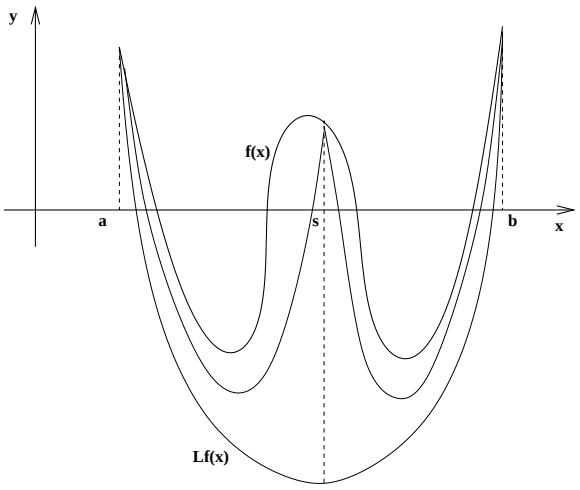

Figure 3: Root finding by Branch and bound method. The mimimum s of the Lower Bound function $Lf(x)$ is used to subdivide the interval of the polynomial $f(x)$.

evaluation when computing lower bound. The recursive algorithm can be given as follows :

5 ILLUSTRATIVE EXAMPLES AND COMPUTATIONAL RESULTS

Ill-conditioned dependance of the zeros on the coefficients occurs for many polynomials having no multiple or clustered zeros, the well known example is the polynomial $\prod_{i=0}^{i=n}(x - i/n)$. For a large n, the zeros jump dramatically because of a smaller perturbation of the coefficients [4]. Furthermore, it would not be appropriate to ignore polynomials with multiple zeros like $(x - 1/2)^n$ since they frequently appear in CAGD. We propose to compare the BR algorithm and the BB algorithm with help of these polynomials. The numerical computations were implemented with the $IEEE754$ double precision floating point arithmetic. The average relative error of polynomial zeros is used, for the comparison.

Let z_1, z_2,..., z_k be the exact polynomial zeros and $\bar{z}_1$, $\bar{z}_2$, ..., $\bar{z}_k$ be the zeros determined with an experimental method. The relative error ξ_i on the zero z_i is determined as follows:

$$\xi_i = Min_{0 \leq j \leq k} \frac{|z_i - \bar{z}_j|}{|z_i|}. \tag{11}$$

Algorithm 2: Branch and Bound algorithm

Function $S = \mathbf{ZeroPolynom}(f, n, a, b, \epsilon)$;

Data: f : the polynomial, n : the degree of the polynomial, a, b : the end of the interval $[a, b]$, ϵ : precision of the zeros

Result: S - the set of all found zeros of f

begin

$S_k = \emptyset$ is an intermediate set;

if $(b - a) < \epsilon$ **then** $S = \emptyset$, **return** S;

;

Else Compute $f(a)$;

if $|f(a)| < \epsilon$ **then** $S_k = S_k \cup \{a\}$;

;

if $f(a) > 0$ **then**

Construct Lf_k, a quadratic lower bound of f on the interval $[a, b]$;

Calculate $x_k^L = minLf_k(x)$ on $[a, b]$;

if $|Lf_k(x_k^L)| < \epsilon$ **then** $S_k = S_k \cup \{x_k^L\}$;

;

else if $Lf_k(x_k^L) > 0$ **then** $S_k = \emptyset$;

;

else

Subdivide the interval $[a, b]$ into $[a, x_k^L]$ and $[x_k^L, b]$;

$S_k = S_k \cup ZeroPolynom(f, n, a, x_k^L, \epsilon)$;

$S_k = S_k \cup ZeroPolynom(f, n, x_k^L, b, \epsilon)$;

end

end

else

Construct Uf_k, a quadratic upper bound of f on the interval $[a, b]$;

Calculate $x_k^U = maxUf_k$ on $[a, b]$;

if $|Uf_k(x_k^U)| < \epsilon$ **then** $S_k = S_k \cup \{x_k^U\}$;

;

else if $Lf_k(x_k^U) < 0$ **then** $S_k = \emptyset$;

;

else

Subdivide the interval $[a, b]$ into $[a, x_k^U]$ and $[x_k^U, b]$;

$S_k = S_k \cup ZeroPolynom(f, n, a, x_k^U, \epsilon)$;

$S_k = S_k \cup ZeroPolynom(f, n, x_k^U, b, \epsilon)$;

end

end

$S = S_k$, **return** S;

end

Table 1: Computed zeros of the polynom P_1 having zeros uniformily distributed.

Zeros of the polynom	Zeros found with BR method	Zeros found with BB method
0.100000000	0.100000000665435	0.100000000315568
0.200000000	0.199999999854139	0.199999999731046
0.300000000	0.300000000538633	0.299999999069959
0.400000000	0.400000000151056	0.400000000461224
0.500000000	0.500000000567752	0.500000000755813
0.600000000	0.600000000255243	0.600000000017924
0.700000000	0.700000000353192	0.700000000555602
0.800000000	0.800000000850357	0.800000000051541
0.900000000	0.899999999124374	0.899999999448404
Relative Error	0.000000001	0.000000001
Time (seconde)	0.020000000	0.000000000

This definition is meaningful, for it takes into account the possible missed zeros. The average relative error is given by:

$$\xi = \frac{1}{n}\sum_{i=1}^{n}\xi_i. \tag{12}$$

1. **Polynomials of the form $\prod_{i=0}^{i=n}(x - i/n)$.**

   The experimental result shows that up to $n = 20$, the proposed algorithm found every zero. Beyond $n = 20$, the method start to fail and the results deteriorate. This is due to successive division operations performed by the algorithm in the power basis. For $n = 9$, we construct the polynomial $P(x)$ by multipliying the monomials $(x - 1/10)\ldots(x - 9/10)$.

   $P_1(x) = -0.000362880 + 0.010265760 * x^1 - 0.117270000 * x^2 + 0.723680000 * x^3 - 2.693250000 * x^4 + 6.327300000 * x^5 - 9.450000000 * x^6 + 8.700000000 * x^7 - 4.500000000 * x^8 + 1.000000000 * x^9$.

   The roots of the polynom $P_1(x)$ are presented in the table 1. The roots are found with an excellent precision (about 10^{-9}).
   As we can see, the BR algorithm is slower than the BB algorithm and the average relative error of the two algorithms are equivalent.

Table 2: Computed zeros of a polynom P_2 having arbitrary zeros in $[0, 1]$.

Zeros of the polynom	Zeros found with BR method	Zeros found with BB method
0.020600000000000	0.020599999759396	0.020600000869723
0.056600000000000	0.056600000103261	0.056600000165920
0.079900000000000	0.079900000770853	0.079899999736290
0.210000000000000	0.210000000484068	0.209999999687340
0.397300000000000	0.397300000026743	0.397299999510668
0.446600000000000	0.446599999948637	0.446600000693173
0.577600000000000	0.577600000509391	0.577599999576455
0.955100000000000	0.955100000384805	0.955099999443358
0.979100000000000	0.979100000180963	0.979099999479073
0.983500000000000	0.983499999757962	0.983500000935180
Relative Error	0.000000003	0.000000006
Time (seconde)	0.020000000	0.000000000

2. **Polynomials of the form $\prod_{i=0}^{i=n}(x - \alpha_i)$ with $0 < \alpha_i < 1$.**

   The numbers α_i are randomly chosen in $[0, 1]$. For $n = 10$, we have the polynom :

   $$P_2(x) = 0.000001844 * x^0 - 0.000171607 * x^1 + 0.005343025 * x^2 - 0.076828641 * x^3 + 0.577913722 * x^4 - 2.479205141 * x^5 + 6.376540019 * x^6 - 9.980342796 * x^7 + 9.283051040 * x^8 - 4.706300000 * x^9 + 1.000000000 * x^{10}$$

   As in the previous experience our methods found all the zeros with high accuracy (about 10^{-9}). This experience (and the previous) shows that the manner in which the zeros are distributed (at random or uniformly) has no influence on the performance of the method. Only the density has an effect on their stability as we can see in the next experience.

3. **Polynomials of the form $\prod_{i=1}^{i=n}(x - 1/2^i)$.**

   For $n = 8$, the polynom is given by:

   $$P_3(x) = 0.000000000 * x^0 + -0.000000007 * x^1 + 0.000001257 * x^2 + -0.000090483 * x^3 + 0.002991959 * x^4 + -0.046327114 * x^5 + 0.329437256 * x^6 + -0.996093750 *$$

Table 3: Computed zeros of the polynom P_3 having zeros $1/2, 1/4, 1/8, ...1./256$

Zeros of the polynom	Zeros found with BR method	Zeros found with BB method
0.500000000000000	0.499999999976857	0.500000000692734
0.250000000000000	0.249999999424596	0.249999999331309
0.125000000000000	0.124999999051953	0.125000000435511
0.062500000000000	0.062500000947371	0.062499999276549
0.031250000000000	0.031249999046515	0.031250000902864
0.015625000000000	0.015625000953089	0.015624999929206
0.007812500000000	0.007812500953198	0.007812500729477
0.003906250000000	0.003906250953166	0.003906250723726
Relative Error	0.000000060	0.000000041
Time (seconde)	0.050000000	0.010000000

$x^7 + 1.000000000 * x^8$.

The zeros of the polynom are presented in the table 3. The two algorithms find all the zeros and confirms the results of the above experiences. As we can see, the average relative error is about 10^{-8}. However the computation time of the two methods increases in comparison with the two above experiences. This is due to the density of the zeros.

4. **Polynomials of the form** $(x - 1/3)^n(x - 1/2)(x - 2/3)$**.**

For these polynomials, the multiplicity n of the value $1/3$ varies from 2 to 9. For any n, the zero $1/3$ is found as a simple zero. For $n = 3, 4, 9$, the results are summarized in the following tables :

For $n = 3$, we have the polynom : $P_4(x) = -0.012345679 * x^0 + 0.154320988 * x^1 - 0.759259259 * x^2 + 1.833333333 * x^3 - 2.166666667 * x^4 + 1.000000000 * x^5$. The zero $1/3$ is found with a good precision (about 10^{-6}).

For n=4, the Polynom is : $P_5(x) = 0.004115226 - 0.063786008 * x^1 + 0.407407407 * x^2 - 1.370370370 * x^3 + 2.555555556 * x^4 - 2.500000000 * x^5 + 1.000000000 * x^6$.

In this case, the BR method misses the zero $1/3$, consequently, the average relative error increases abnormally. The BB method finds this zero twice. The

Table 4: Zeros of the polynom P_4 with BR and BB algorithm. The multiplicity of 1/3 is $n = 3$

Multiplicity of 1/3	Zeros found with BR method	Zeros found with BB method.
n=3	0.333329111317813 0.500000000550211 0.666666666945224	0.333332132529702 0.499999999572852 0.666666666134202
Relative Error	0.000007600	0.000002162
Time (seconde)	0.040000000	0.010000000

Table 5: Zeros of the polynom P_5 with BR and BB algorithm. The multiplicity of 1/3 is $n = 4$

Multiplicity of 1/3	Zeros found with BR method	Zeros found with BB method
n=4	0.500000000813783 0.666666666228435	0.333298858981134 0.333603393113499 0.499999999585050 0.666666665802389
Relative Error	0.333333335	0.000068949
Time (seconde)	0.020000000	0.000000000

average relative error is about 10^{-5}.

For $n = 9$, we have the polynom : $P_6(x) = -0.000016935 * x + 0.000516520 * x^1 - 0.007138140 * x^2 + 0.058984911 * x^3 - 0.323731139 * x^4 + 1.238683128 * x^5 - 3.370370370 * x^6 + 6.518518519 * x^7 - 8.777777778 * x^8 + 7.833333333 * x^9 - 4.166666667 * x^{10} + 1.000000000 * x^{11}$.

The zero 1/3 is found by the two methods as the graph of the polynom f crosses the x-axis. The average relative error of the two methods is about 10^{-2}.

5. **Zeros of Laguerre Polynomials.**
For $\alpha \geq 0$, the generalized Laguerre polynomials of order n, denoted by L_n^α, are given by the recurrence

$$(n+1)L_{n+1}^\alpha(x) = (2n + \alpha + 1 - x)L_n^\alpha(x) - (n + \alpha)L_{n-1}^\alpha(x) \qquad (13)$$

Table 6: Zeros of the polynom P_6 with BR and BB algorithm. The multiplicity of 1/3 is $n = 9$

Multiplicity of 1/3	Zeros found with BR method	Zeros found with BB method
n=9	0.319624464534332 0.500000000812627 0.666666667070851	0.343752181314919 0.500000000335154 0.666666667207756
Relative Error	0.033649042	0.025573536
Time (seconde)	0.020000000	0.000000000

Table 7: Computed zeros of the Laguerre polynom of degree $n = 10$

Zeros of the polynom	Zeros found with BR method	Zeros found with BB method
0.137793470540	0.137793731205827	0.137793487093318
0.729454549503	0.729453277475226	0.729454566889281
1.808342901740	1.808342741998582	1.808342880722087
3.401433697855	3.401433565079064	3.401433705686546
5.552496140064	5.552497088211168	5.552496160310486
8.330152746764	8.330152247669790	8.330152735747728
11.843785837900	11.843783590823241	11.843785821366534
16.279257831378	16.279257793680930	16.279257837310475
21.996585811981	21.996585929553266	21.996585792878673
29.920697012274	29.920697472096251	29.920697001378144
Relative Error	0.000000042	0.000000017
Time (seconde)	0.010000000	0.010000000

and the initial condition $L_0^\alpha(x) = 1 \quad L_1^\alpha(x) = 1 + \alpha - x$. It is well known that $L_n^\alpha(x)$ has n real and simple zeros, which are all in the interval $(0, n + \alpha + (n-1)\sqrt{n+\alpha}]$. The generalized Laguerre polynomials satisfy the monotonocity and interlacing properties of zeros which means that zeros of $L_{n-1}^\alpha(x)$ and zeros of $L_n^\alpha(x)$ separate each other. This propertie may be used for the approximation of the zeros of the Laguerre polynomials by a fixed point iteration. Table 7 summarize the result of the computed zeros for the Laguerre polynomials of order $n = 10$ and $\alpha = 0$. In [16], the first fifteen zeros of Laguerre polynomials are given.

6 CONCLUSION

We propose two approaches for finding all real zeros of a polynomial $f(x)$ of degree n. The methods are based on the Branch and Bound method by the computation of some lower and upper bounds of $f(x)$ and on successive reduction or subdivision of the initial interval. Facing ill-conditionned polynomial, the experimental results show the efficiency of the two algorithms. Simple zeros are found with good accuracy (relative error magnitude $= 10^{-9}$). Multiple zeros can be found as simple zero. In the multiple zeros case, the BB algorithm is more efficient than the BR algorithm. For high multiplicity, the two algorithms provide bad results, though the BB algorithm behaves a bit better than the BR algorithm. The average relative error in this case is about 10^{-2}. As the computations are performed in a finite precision arithmetic and rounding errors affect the coefficients of polynomials of high degree, our results deteriorate beyond $n = 20$. But, it has been shown [5, 11] that the Bernstein basis minimizes the condition number which measures the sensibility of the zeros through the coefficients perturbation. Our target is to use this base to improve the stability of the proposed algorithm.

REFERENCES

[1] Pan, V.Y., "Solving a polynomial equation: some history and recent progress", *SIAM Review*, 39 (1997)187–220.

[2] Jenkis, M.A., Traub, J.F., "A three-stage algorithm for real polynomials using quadratic iteration", *SIAM Journal on Numerical Analysis*, 7 (1970)545–566.

[3] Mourrain, B., Vrahatis M.N., Yakoubsohn, J.C., "On the complexity of isolating real roots and computing with certainty the topological degree", *Journal of Complexity*, 18 (2002)612–640.

[4] Wilkinson, J.H., "The evaluation of the zeros of ill-conditioned polynomials Part I", *Numer Math.*, 1 (1959) 150–166.

[5] Farouki, R.T., Rajan, V.T., "Algorithms for polynomials in Bernstein form", *International journal of Computer Aided Geometric Design*, 5 (1988)1–26.

[6] De Boor, C., *A practical Guide to Splines Applied Mathematical Sciences*, Springer Verlag, 1978.

[7] Zidna, A., Michel, D., "A two-steps algorithm for approximating real roots of a polynomial in Bernstein Basis", *International Journal of Mathematics and Computers in Simulation*, 77(2008)313–323.

[8] Falk, J.E, Soland, R.M., "An algorithm for separable nonconvex programming problems", *Manage. Sci*, 15 (1969)550–569.

[9] Le Thi, H.A., Ouanes, M., "Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint", *Rairo-Operations Research*, 40 (2006)285–302.

[10] Le Thi, H.A., Ouanes, M., Zidna, A., "An adapted Branch and Bound Algorithm for approximating real roots of a polynomial", *MCO*, (2008)182-189.

[11] Kiciak, P., Zidna, A., "Recursive de Casteljau bisection and rounding errors", *Journal of computer Aided Geometric Design*, 21 (2004)683–695.

[12] Calvin, J., Ilinskas, A., "On the convergence of the P-algorithm for one-dimensional global optimization of smooth functions", *Journal of Optimization Theory and Applications*, 102 (1999)479–495.

[13] Hansen, P., Jaumard, B., Xiong, J., "Decomposition and interval arithmetic applied to minimization of polynomial and rational functions", *Journal of Global Optimization*, 3 (1993)421–437.

[14] Hansen, P., Jaumard, B., Lu, S.H., "Global Optimization of Univariate Functions by Sequential Polynomial Approximation", *International Journal of Computer Mathematics*, 28(1989)183–193.

[15] Hansen, P., B., Jaumard, B., Lu, S.H., "Global Optimization of Univariate Lipschitz Functions: 2. New Algorithms and Computational Comparison", *Mathematical Programming*, 55(1992)273–292

[16] Salzer, H., E., Zucker, R., "Table of the zeros and weight factors of the first fifteen Laguerre polynomials", *Bulletin of the American Mathematical Society*, 55 (1949)1004–1012.

[17] Moore, R., *Interval Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[18] Thai, Q.P., Le Thi H.A., Pham, D.T., "On the global solution of linearly constrained indefinite quadratic minimization problems by decomposition branch and bound method", *RAIRO, Recherche Opérationnelle*, 30(1996) 31–49.

[19] Ratschek, H., Rokne, J., *New Computer Methods for Global Optimization*, Wiley, New York, 1982.

[20] Ratz, D., "A nonsmooth global optimization technique using slopes the one-dimensional case", *Journal of Global Optimization*, 14 (1999)365–393.

[21] Visweswaran, V., Floudas, C.A., *Global Optimization of Problems with Polynomial Functions in One Variable*, in: Recent Advances in Global Optimization, pp. 165–199. Floudas A. and P.M Pardalos P.M. (ed.), Princeton University Press, Princeton, 1992.

[22] Sergeyev, Ya.D., "Global one-dimensional optimization using smooth auxiliary functions", *Mathematical Programming*, 81 (1998)127–146.