NUMERICAL ALGEBRA,
CONTROL AND OPTIMIZATION
Volume **11**, Number **1**, March **2021**

doi:10.3934/naco.2020015

pp. **63–73**

PIECEWISE QUADRATIC BOUNDING FUNCTIONS FOR FINDING REAL ROOTS OF POLYNOMIALS

Djamel Aaid*

Department of the Common Base Natural and Life Sciences
Batna 2 University, Algeria

Amel Noui

Faculty of Mathematics and Computer Science
Batna 2 University, Algeria

Özen Özer

Department of Mathematics, Faculty of Science and Arts
Kirklareli University, 39100, Kirklareli, Turkey

(Communicated by Yongzhong Song)

Abstract. In this paper, our main interest is to create/ construct a new useful and outstanding algorithm to obtain roots of the real polynomial represented by $f(x) = c_0 + c_1x + ... + c_ix^i + ... + c_nx^n$ where coefficients of the polynomials are real numbers and x is a real number in the closed interval of $\mathbb{R}$. Also, our results are supported by numerical examples. Then, a new algorithm is compared with the others (writer classical methods) and this algorithm is more useful than others.

Introduction. In the third millennium BC, solutions of the $f(x) = c_0 + c_1x + ... + c_ix^i + ... + c_nx^n$ polynomial was initially determined. Still, some open problems on this kind of polynomial are unsolved even if many important and useful results were obtained on such polynomial. There are many methods such as Bairstow's Method, Bernoulli's Method, Graee's Root-squaring Method, Laguerre's Method, Eigen- values of Companion Matrix etc., which are used to find roots of real polynomials.

In [1, 8, 7], D. Aaid et al. proposed an approximation to obtain a global minimum of the univariate objective function. C. S. Adjiman and his co-authors created/constructed a based Branch and Bound global optimization useful on the bounds of global minimum in [2]. Chen, X.D et al. gave some kind of clipping method to compute roots of polynomials in [5, 3, 4]. Also, the book of C. De Boor [6] includes different type of methods to calculate the root of polynomials. In [10] Le Thi and his collaborators approached (P) and (PC) problems. They created an algorithm with numerical examples for then by working as Branch and Bound approximations. Then, they described an adapted Branch and Bound algorithm to approximate the real roots of a polynomial in [11]. Some potential authors with A. Zidna also

2020 *Mathematics Subject Classification.* 26C10, 90C20, 90C25, 90C90.

Key words and phrases. Global optimization, Piecewise Quadratic Upper Function, Piecewise Quadratic Lower Function, Root-finding, Branch and Reduce, Branch and Bound.

* Corresponding author: Djamel Aaid.

proposed two algorithms to obtain all zeros of a polynomial in a power basis (BR) and (BB) in [12].

In [13], M. Ouanes et al. demonstrated new quadratic lower bound and supported their results by numerical examples. SLEFE theory was introduced and determined SLEFE isolation algorithm by P. Jiang et al. in [9]. A. Shpak [14] constructed a significant theory on a wider class of objective functions.

The main problem is to solve the real polynomial equation:

$$f(x) = c_0 + c_1 x + ... + c_i x^i + ... + c_n x^n = 0 \tag{1}$$

with $n \geq 2$ for $c_i \in \mathbb{R}$ $(\forall i = 0, ..., n)$ and $x \in [a, b] \subset \mathbb{R}$.

In this work, we propose an efficient combination among Branch, Bound and Reduce method with piecewise tighter bounds see Figure (1−3) to compute real roots of real polynomials.

We construct piecewise quadratic underestimation and piecewise overestimation functions for a given polynomial f over a closed interval $[a, b]$ with some properties.

Benefits of the paper can be given as follows:

1. The polynomial is well framed since the distance between the polynomial and the bounds functions is reduced as long as we need/want.
2. The concavity and convexity test detects the regions whether or not the function is convex/concave.
3. To remove subintervals, which don't include zeros of a polynomial, make process speedy and increase the rate of reduction.

Advantages of this method can be said that the polynomial is well framed and the distance can be small as one may like it, which makes the process is faster. The results on the piecewise quadratic lower bounding function are better and more useful than other techniques such as given in [6], [10] or [2].

Paper is prepared by using four sections such as preliminaries, algorithms, proofs, numerical examples, etc.

1. **Piecewise Quadratic Bounds.** Considering definition of the lower bound, the upper bound was constructed in [1] similarly. Assume that $X = [a, b]$ be a bounded closed interval in $\mathbb{R}$ and f be a continuous differentiable polynomial with second degree on X. For x^0 and x^1 real numbers in $[a, b]$ such that $x^0 \leq x^1$, l_0 and l_1 real valued functions were defined in [6] as follows:

$$\begin{cases} l_0(x) = \frac{x^1 - x}{x^1 - x^0} & \text{if } \ x^0 \leq \ x \leq x^1 \\ l_1(x) = \frac{x - x^0}{x^1 - x^0} & \text{if } \ x^0 \leq \ x \leq x^1 \end{cases} . \tag{2}$$

Also, we have the following equations:

$$l_0(x) + l_1(x) = 1, \ \forall x \in [a, b], \tag{3}$$

and

$$l_i(x^j) = \begin{cases} 0 \ ; \ i \neq i \\ 1 \ ; \ i = j \end{cases} . \tag{4}$$

We suppose that $h = x^1 - x^0$ and $L_h f$ be the linear interpolant to f at points x^0, x^1. Then, the following equation holds where $f(x)$ is a univariate polynomial with degree $n > 1$.

$$L_h f(x) \ = \sum_{i=0}^{1} l_i(x) f(x^i). \tag{5}$$

Let us find the whole roots of it in the closed interval $[a,b]$, which is divided into the n equal subintervals, for $x_i = a + ih$, $h = \frac{b-a}{n}$ and $i = 0, ..., n$.

Then, we can create a new overestimator of f which is corresponding with local quadratic overestimation as:

$$U_i(x) = L_{h_i} f(x) + Q_i(x),\ i = 0, ..., n-1, \tag{6}$$

where $Q_i(x) = \frac{1}{2} K_i (x - x_i)(x_{i+1} - x)$ for $i = 0, ..., n-1$. In this equation, K_i (which is valid for $[x_i, x_{i+1}]$) is defined by upper bound of the second derivation. Our aim is to construct a piecewise quadratic upper bound instead of quadratic upper bound over $[a,b]$.

Theorem 1.1. *Let f be a function as follows*

$$f(x) \leq U(x) \text{ for all } x \in [a,b],$$

such that

$$U(x) = \begin{cases} U_0(x)\ ;\ x \in [x_0, x_1] \\ \quad ... \\ U_i(x);\ x \in [x_i, x_{i+1}] \\ \quad ... \\ U_{n-1}(x);\ x \in [x_{n-1}, x_n] \end{cases} . \tag{7}$$

Then, $U(x)$ is a continuous piecewise concave over-estimator of $f(x)$ for all x in $[a,b]$.

Proof. We consider the ϕ function (which is defined on closed subintervals $[x_i, x_{i+1}]$ for $i = 0, ..., n-1$) as follows:

$$\phi(x) = U_i(x) - f(x) = L_{h_i} f(x) + \frac{1}{2} K_i (x - x_i)(x_{i+1} - x) - f(x). \tag{8}$$

Considering (7) equation, we get the following equation:

$$\phi''(x) = -f''(x) - K_i \leq 0,\ i = 0, ..., n-1,\ x \in [x_i, x_{i+1}].$$

This shows that ϕ is a concave function and we obtain that following inequality is satisfied

$$\phi(x) \geq \min\{\phi(x),\ x \in [x_i, x_{i+1}]\} = \phi(x_i) = \phi(x_{i+1}) = 0. \tag{9}$$

□

1.1. **Concave convex test.** α_i and β_i $(i = 0, ..., n-1)$ values can be computed on each subintervals $[x_i, x_{i+1}]$, $(i = 0, ..., n-1)$ where :

$$\alpha_i \leq f''(x) \leq \beta_i, \text{for all } x\ \in [x_i, x_{i+1}].$$

Besides, the following statements are obtained for the test.

- if $\alpha_i \geq 0$ (it means that $0 \leq f''(x)$ is satisfied for all $x \in [x_i, x_{i+1}]$, $(i = 0, ..., n-1)$ then f is a convex function on the closed subinterval $[x_i, x_{i+1}]$. So, the local lower bound holds $L_i f(x) = f(x)$ on the closed interval $[x_i, x_{i+1}]$.
- if $\beta_i \leq 0$ (it means that $f''(x) \leq 0$ is held for all $x \in [x_i, x_{i+1}]$, $(i = 0, ..., n-1)$ then f is a concave function on the closed subinterval $[x_i, x_{i+1}]$, $(i = 0, ..., n-1)$. Thus, the local upper bound $U_i f(x) = f(x)$ satisfies on $[x_i, x_{i+1}]$, $(i = 0, ..., n-1)$.

So, we can easily say that the β_i and α_i, $(i = 0, ..., n-1)$ are computed by interval analysis (for background and more information, one may look at the reference [14]).

Remark 1. Following properties are satisfied by proposed bounds:

1. If the concave convex test is satisfied, then the proposed bounds match up with the $f(x)$ polynomial.
2. The lower bound is continuous piecewise convex on the closed interval $[a, b]$.
3. The upper bound is continuous piecewise concave on the closed interval $[a, b]$.
4. Both lower bound and upper bound match up with the $f(x)$ function at the end point of the closed subinterval $[x_i, x_{i+1}]$ for $i = 0, ..., n-1$.
5. Both lower bound function and upper bound function are determined explicitly.
6. The upper bound and lower bound functions would approach the polynomial as much as we would like.

2. **Branch, Bound And Reduce Algorithm (BBR).** In this section, we describe the Branch, Bound and Reduce algorithm for the sake of finding/obtaining all real roots of the polynomials where the polynomial degree is greater than one. To get all zeros of the polynomial with degree $n \geq 2$, we need to determine subdivision of the closed interval $[a, b]$ in n part like $[x_i, x_{i+1}]$ for $i = 0, ..., n-1$.

We know that f polynomial has n roots which are taken into consideration the multiplication of zeros. This result includes two different cases defined as follows;

1. In the first case: if $f(x_i) > 0$, then we are interested in the local lower bound by solving the equation on each subinterval $[x_i, x_{i+1}]$ with second degree. There are three (3) different conditions to investigate for
$$L_i f(x_i) = 0$$
as follows;
   - Conditions 1: if there is no zero of the local lower bound equation
   $$L_i f(x_i) = 0$$
   for some $i = 0, ..., n-1$, then the interval $[x_i, x_{i+1}]$ will not be considered for algorithm.
   - Conditions 2: if the equation
   $$L_i f(x_i) = 0$$
   has just one solution r, then we check whether or not the latter is a zero of polynomial f and look for more precise searches by reducing this part to $[r + \varepsilon, x_{i+1}]$.
   - Conditions 3: if the equation
   $$L_i f(x_i) = 0$$
   has two different roots r_1 and r_2, then we have to check whether the latter are a zeros of polynomial f or searching in the reducing part $[r_1+\varepsilon, r_2-\varepsilon]$.
2. In the second case : if $f(x_i) < 0$, then one quit at the local upper bound by solving the equation of the second degree on $[x_i, x_{i+1}]$.
$$U_i f(x_i) = 0.$$

In a similar way, if we proceed with all steps of the first case, then the following algorithm can be given.

Algorithm

Input :

- $[a, b]$: a real interval.

- ε : the accuracy.
- f: the polynomial.
- n: the degree of the polynomial.

Output :

- Z: the set of all found zeros of f (Z=ZeroPolynom(f, n, a, b)).

Subdivision and evaluation

for all $i = 0, ..., n$

- Compute $x_i = a + \frac{b-a}{n} i$, and set $M = \bigcup_{i=0}^{n-1} \{[x_i, x_{i+1}]\}$
- **while** $\exists\ i$ such that $(x_{i+1} - x_i) \geq \varepsilon$ **do**

for all $i = 1, ..., nb$ such as nb: number of subintervals in M

- Compute β_i, α_i and K_i on each $[x_i, x_{i+1}]$
- Compute $f(x_i)$

First case $(f(x_i) > 0)$

- Solve the equation $L_i f(x) = 0$;

1. **if** $L_i f(x)$ has no root in $[x_i, x_{i+1}]$ **then** delete $[x_i, x_{i+1}]$ to the set M
2. **else if** $L_i f(x)$ has one root $r \in [x_i, x_{i+1}]$
   - **if** $|f(r)| < \varepsilon$, **then** $Z_i = Z_i \cup \{r\}$,
   - $Z_i = Z_i \cup$ ZeroPolynom$(f, n, r + \varepsilon, x_{i+1})$
3. **else if** $L_i f(x)$ has two roots r_1 and r_2 in $[x_i, x_{i+1}]$, **then**
   - **if** $|f(r_1)| < \varepsilon$, **then** $Z_i = Z_i \cup \{r_1\}$
   - **if** $|f(r_2)| < \varepsilon$, **then** $Z_i = Z_i \cup \{r_2\}$
   - $Z_i = Z_i \cup$ ZeroPolynom$(f, n, r_1 + \varepsilon, r_2 - \varepsilon)$

Second case $(f(x_i) < 0)$

- Solve the equation $U_i f(x) = 0$;

1. **if** $U_i f(x)$ has no root in $[x_i, x_{i+1}]$ **then** delete $[x_i, x_{i+1}]$ to the set M
2. **else if** $U_i f(x)$ has one root $r \in [x_i, x_{i+1}]$,
   - **if** $|f(r)| < \varepsilon$, **then** $Z_i = Z_i \cup \{r\}$,
   - $Z_i = Z_i \cup$ ZeroPolynom$(f, n, r + \varepsilon, x_{i+1})$
3. **else if** $U_i f(x)$ has two roots r_1 and r_2 in $[x_i, x_{i+1}]$, **then**
   - **if** $|f(r_1)| < \varepsilon$, **then** $Z_i = Z_i \cup \{r_1\}$
   - **if** $|f(r_2)| < \varepsilon$, **then** $Z_i = Z_i \cup \{r_2\}$,
   - $Z_i = Z_i \cup$ ZeroPolynom$(f, n, r_1 + \varepsilon, r_2 - \varepsilon)$

$Z = Z_i$, **Return** Z

end algorithm.

Remark 2. The proposed BBR algorithm is specially designed to deal with polynomials, neither convex nor concave. By using the concave convex test, sometimes we come across partitions in which the polynomial is convex or concave. In this case, we will not need to go through all the steps of the algorithm, just solve the equation $U_i f(x) = 0$ or $L_i f(x) = 0$. If we come across a partition in which the polynomial is neither convex nor concave, it means the concave convex test is not verified. Consequently, we will have to frame it by our bounding functions by applying the procedures of the BBR algorithm step by step until all the zeros of the polynomial are obtained.

Illustrative example

Assume that $f(x) = x(x-1)(x-2) = 0$ polynomial in $[0, 2]$. For finding roots of polynomial, we apply our algorithm to the polynomial.

First, we divide the closed interval $[0,2]$ into the two equal parts such as $[0,1]$ and $[1,2]$. If we apply our method (concave convex test) on to f polynomial, we get $f''([0,1]) \leq 0$ implies that f is concave on $[0,1]$ and also $\{0,1\}$ is the set of roots of f due to the solutions of $U_1(x) = 0$. So, we obtain that the interval $[0,1]$ will be removed from the search list. Besides, we have $f''([1,2]) \geq 0$. This gives that f is convex on $[1,2]$ and $\{1,2\}$ is the set of roots of f because of the solutions of $L_2(x) = 0$. Therefore, the polynomial has only three roots as 0,1 and 2. So, we can be able to determine the polynomial's roots, thanks to concave convex test.

2.1. Converge of the Algorithm.

Theorem 2.1. *The algorithm can be stopped if the least one of the following conditions is satisfied.*

1. *The length of all closed subinterval $[x_i, x_{i+1}]$ is less than ε.*
2. *The local lower bound (local upper bound) of the polynomial has no root in all closed subinterval $[x_i, x_{i+1}]$ for $\varepsilon = 0$. So, we get $\lim_{h_i \to 0}(U_i f(x) - f(x)) = 0$ and $\lim_{h_i \to 0}(f(x) - L_i f(x)) = 0$ for $(h_i = x_{i+1} - x_i\ )$*

Proof. We can prove that

$$\lim_{h_i \to 0}(U_i f(x) - f(x)) = 0,$$

since we have the following inequalities

$$0 \leq U_i f(x) - f(x) \leq \frac{1}{2}K_i(x - x_i)(x_{i+1} - x) \leq \frac{1}{2}K_i h_i^2.$$

In a similar way,

$$\lim_{h_i \to 0}(f(x) - L_i f(x)) = 0,$$

due to inequality

$$0 \leq f(x) - L_i f(x)(x) \leq \frac{1}{2}K_i(x - x_i)(x_{i+1} - x) \leq \frac{1}{2}K_i h_i^2.$$

□

3. Numerical Examples.

To measure the performance of our BBR algorithm, we compare our algorithm with BB algorithm given in [12] and BR algorithm presented in [11, 12]. BB and BR algorithms are implemented by C^{++} programme language with double precision floating point. Also, the property of computer is Intel (R) Core (TM) i3-311MCP4 with CPU 2.40GHz. Numerical results are given within four examples.

In this work, our aim is to compute α_i, β_i and K_i constants. So, we consider the combination of BB and BR methods in our method BBR.

Here below, we give four examples and we compare our results with those of published papers [11, 12]. We use approximate computations with a small precision 10^{-9} for the idea of our paper (our precision). Let z_1, z_2, ..., z_k be the exact polynomial zeros and $r_1, r_2, ..., r_k$ be the zeros determined with an experimental method. The relative error ζ_i on the zero z_i is determined as follows:

$$\zeta_i = \min_{0 \leq i \leq k} \frac{|z_i - r_i|}{|z_i|}.$$

The average relative error is given by:

$$\zeta = \frac{1}{n}\sum_{i=1}^{n}\zeta_i.$$

Our proposed method BBR uses piecewise bounding functions that require fine subdivision of the interval to be explored, involving a multiple research that can detect all cases, namely; convex (concave) regions, neither convex nor concave regions, and regions that do not contain zeros of the polynomial. By pressing this technique, the BBR method will determine all the zeros of the polynomial with an excellent accuracy 10^{-9} and with the least relative error in a short time.

Example 1

Let $P(x) = \prod_{i=0}^{n}\left(x - \frac{i}{n}\right)$ be a polynomial. If it is given by multiplying the monomials $(x - 1/10)...(x - 9/10)$, then we obtain

$$\begin{aligned} P_1(x) = &-0.000362880 + 0.010265760x - 0.117270000x^2 + 0.723680000x^3 \\ &- 2.693250000x^4 + 6.327300000x^5 - 9.450000000x^6 + 8.700000000\ x^7 \\ &- 4.500000000x^8 + 1.000000000x^9. \end{aligned}$$

TABLE 1. Numerical results of Example 1

Zeros of polynom	Zeros found with BR	Zeros found with BB	Zeros found with BBR
0.100000000	0.100000000	0.100000000	0.100000000
0.200000000	0.199999999	0.199999999	0.199999999
0.300000000	0.300000000	0.299999999	0.300000000
0.400000000	0.400000000	0.400000000	0.400000000
0.500000000	0.500000000	0.500000000	0.500000000
0.600000000	0.600000000	0.600000000	0.600000000
0.700000000	0.700000000	0.700000000	0.700000000
0.800000000	0.800000000	0.800000000	0.800000000
0.900000000	0.899999999	0.899999999	0.900000000
Relative Error	0.000000002	0.000000003	0.000000001
number of iterations	25	22	06
Time (second)	0.082000000	0.010000000	0.000000000

Example 2

Assume that the polynomial $P(x)$ is defined by $P(x) = \prod_{i=0}^{n}(x - \theta_i)\ \ 0 < \ \theta_i < 1$. If we chose $n = 10$, then we get the following polynomial:

$$\begin{aligned} P_2(x) = &\ 0.000001844x0 - 0.000171607x + 0.005343025 * x^2 - 0.076828641x^3 \\ &+ 0.577913722x^4 - 2.479205141x^5 + 6.376540019x^6 - 9.980342796x^7 \\ &+ 9.283051040x^8 - 4.706300000x^9 + 1.000000000x^{10}. \end{aligned}$$

Example 3

Let us consider polynomials which are in the form of $\prod_{i=0}^{n}\left(x - \frac{1}{2^i}\right)$.

Table 2. Numerical results of Example 2

Zeros of polynom	Zeros found with BR	Zeros found with BB	Zeros found with BBR
0.020600000	0.020599999	0.020600000	0.020600000
0.056600000	0.056600000	0.056600000	0.056600000
0.079900000	0.079900000	0.079899999	0.079899999
0.210000000	0.210000000	0.209999999	0.210000000
0.397300000	0.397300000	0.397299999	0.397300000
0.446600000	0.446599999	0.446600000	0.446599999
0.577600000	0.577600000	0.577599999	0.577600000
0.955100000	0.955100000	0.955099999	0.955100000
0.979100000	0.979100000	0.979099999	0.979100000
0.983500000	0.983499999	0.983500000	0.983500000
Relative Error	0.000000003	0.000000006	0.000000002
number of iterations	27	32	12
Time (second)	0.076100000	0.052000000	0.016000000

Table 3. Numerical results of Example 3

Zeros of polynom	Zeros found with BR	Zeros found with BB	Zeros found with BBR
0.500000000	0.499999999	0.500000000	0.500000000
0.250000000	0.249999999	0.249999999	0.250000000
0.125000000	0.124999999	0.125000000	0.124999999
0.062500000	0.062500000	0.062499999	0.062500000
0.031250000	0.031249999	0.031250000	0.031249999
0.015625000	0.015625000	0.015624999	0.015624999
0.007812500	0.007812500	0.007812500	0.007812500
0.003906250	0.003906250	0.003906250	0.003906250
Relative Error	0.000000004	0.000000003	0.000000003
number of iterations	36	27	11
Time (second)	0.096200000	0.027300000	0.036100000

For $n = 8$, the polynomial is obtained by:

$$
\begin{aligned}
P_3(x) &= 0.000000000 - 0.000000007x + 0.000001257x^2 + -0.000090483x^3 \\
&\quad + 0.002991959x^4 + -0.046327114x^5 + 0.329437256x^6 - 0.996093750x^7 \\
&\quad + 1.000000000x^8.
\end{aligned}
$$

Example 4

Laguerre Polynomials. For $\theta \geq 0$, the generalized Laguerre polynomials of order n, which is represented by L_n^θ, are given by the following recurrence:

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

for $L_0^\theta(x) = 1$ and $L_1^\theta(x) = 1 + \theta - x$.

We have proposed an approximation to get an automated construction of bounded functions with one variable. The synthesis of bounds are driven by rules applied to algebraic expression of a function. The proposed approach is experimentally compared with adapted (BB) or (BR). Experiments have demonstrated that some functions with proposed method can significantly outperform to standard approaches. It should be noted that our approach can be helpful in a different kind of multivariate functions.

TABLE 4. Numerical results of Example 4

Zeros of polynom	Zeros found with BR	Zeros found with BB	Zeros found with BBR
0.137793470	0.137793487	0.137793487	0.137793486
0.729454549	0.729454566	0.729454566	0.729454565
1.808342901	1.808342880	1.808342880	1.808342881
3.401433697	3.401433705	3.401433705	3.401433705
5.552496140	5.552496161	5.552496160	5.552496160
8.330152746	8.330152734	8.330152735	8.330152735
11.843785837	11.843785821	11.843785821	11.843785821
16.279257831	16.279257837	16.279257837	16.279257837
21.996585811	21.996585793	21.996585792	21.996585792
29.920697012	29.920697000	29.920697001	29.920697002
Relative Error	0.000000001	0.000000001	0.000000001
number of iterations	23	13	07
Time (second)	0.001200000	0.020000000	0.000000000

4. **Conclusion.** It is conspicuous to discern that our work/algorithm is more useful, effective and practical than the classical ones. Indeed, this new algorithm facilitates the determination of the solutions on either convex or concave region due to the solutions of second-degree equation, as well as quadratic lower bound or quadratic upper bound. Also, this algorithm enables us to expedite the process and to eliminate subintervals that do not contain solutions. Finally, we recommend other researchers to use the obtained results in further studies.

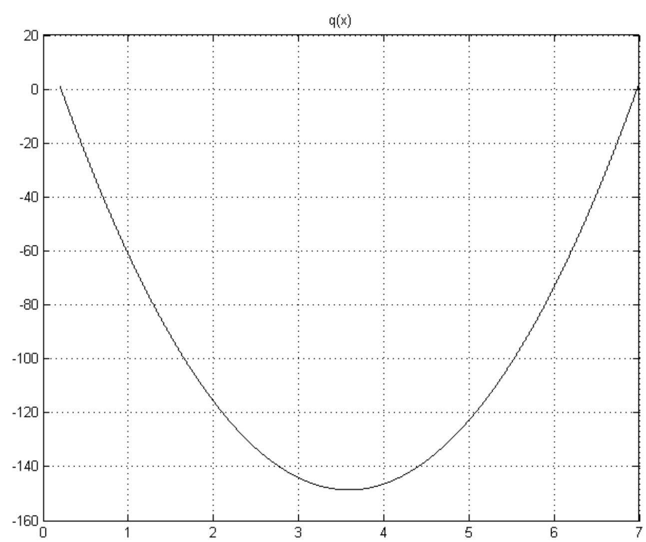

FIGURE 1. The underestimator in (BB) and (BR)

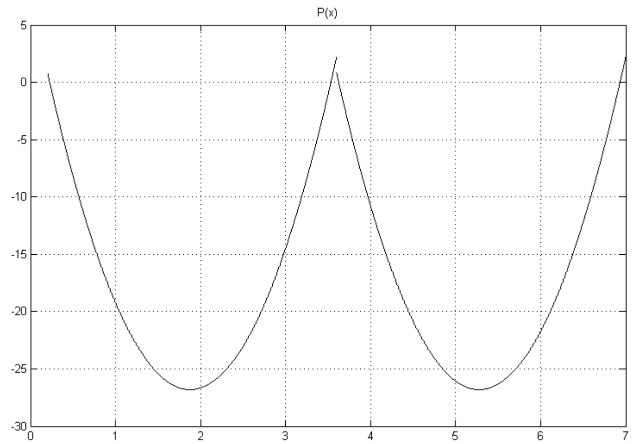

FIGURE 2. The underestimator in our method (BBR)

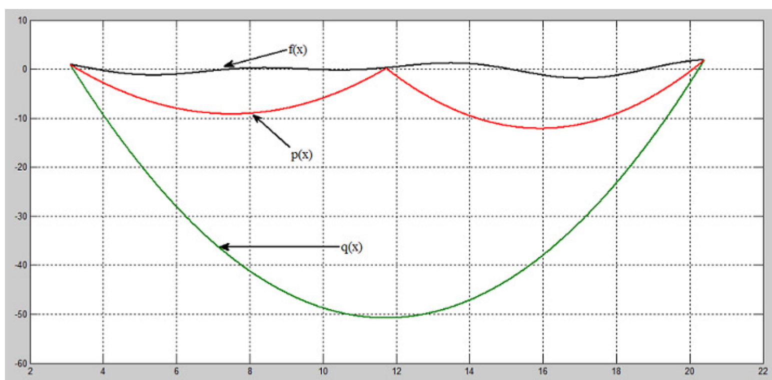

FIGURE 3. Tightness of our underestimator $p(x)$ than the $q(x)$ used in the classical methods (BB) and (BR)

REFERENCES

[1] D. Aaid, A. Noui and M. Ouanes, New technique for solving univariate global optimization, *Archivum Mathematicum*, **53** (2017), 19–33.

[2] C. S. Adjiman, I. P. Androulakis and C. A. Floudas, A global optimization method, αbb, for general twice-differentiable constrained nlpsii. implementation and computational results, *Computers & Chemical Engineering*, **22** (1998), 1159–1179.

[3] X.-D. Chen and W. Ma, A planar quadratic clipping method for computing a root of a polynomial in an interval, *Computers & Graphics*, **46** (2015), 89–98.

[4] X.-D. Chen and W. Ma, Rational cubic clipping with linear complexity for computing roots of polynomials, *Applied Mathematics and Computation*, **273** (2016), 1051–1058.

[5] X.-D. Chen, W. Ma and Y. Ye, A rational cubic clipping method for computing real roots of a polynomial, *Computer Aided Geometric Design*, **38** (2015), 40–50.

[6] C. De Boor, Applied mathematical sciences, in *A Practical Guide To Splines*, Vol. 27, (1978), Spriger-Verlag.

[7] A. Djamel, N. Amel, Z. Ahmed, O. Mohand and L. T. H. An, A quadratic branch and bound with alienor method for global optimization, in *XII Global Optimization Workshop*, (2014), 41–44.

[8] A. Djamel, N. Amel and O. Mohand, A piecewise quadratic underestimation for global optimization, in *The Abstract Book*, (2013), 138.

[9] P. Jiang, X. Wu and Z. Liu, Polynomials root-finding using a slefe-based clipping method, *Communications in Mathematics and Statistics*, **4** (2016), 311–322.

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

[11] H. A. Le Thi, M. Ouanes and A. Zidna, An adapted branch and bound algorithm for approximating real root of a ploynomial, in *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences*, Springer, (2008), 182–189.

[12] H. A. Le Thi, M. Ouanes and A. Zidna, Computing real zeros of a polynomial by branch and bound and branch and reduce algorithms, *Yugoslav Journal of Operations Research*, **24** (2014), 53–69.

[13] M. Ouanes, H. A. Le Thi, T. P. Nguyen and A. Zidna, New quadratic lower bound for multivariate functions in global optimization, *Mathematics and Computers in Simulation*, **109** (2015), 197–211.

[14] A. Shpak, Global optimization in one-dimensional case using analytically defined derivatives of objective function, *The Computer Science Journal of Moldova*, **3** (1995), 168–184.

Received June 2019; 1^{st} revision September 2019; Final revision September 2019.

E-mail address: djamelaaid@gmail.com, d.aaid@univ-batna2.dz
E-mail address: a.noui@univ-batna2.dz
E-mail address: ozenozer39@gmail.com