ISSN: 2633-4828 Vol. 2 No.2, December, 2020

International Journal of Applied Engineering & Technology

B-SPLINE KRAWCZYK POLYNOMIAL SYSTEM SOLVER

Deepak Gawali
Systems & Control Engineering Department, Indian Institute Technology, Bombay
ddgawali2002@gmail.com

ABSTRACT

An effective computational technique for locating all roots of objective polynomial equation with v variables that fall within a y dimensional box is needed for applications in the field of engineering, such as analysis of electrical networks and computer-aided design. We suggest the following method as the basis for our approach to find the polynomial system's roots: The original algebraic equations that were nonlinear are first transformed into polynomial B-spline form, and then a pruning step utilizing the B-spline Krawczyk operator is included.

In this study, we conduct a comparative analysis of the performance of the suggested B-spline Krawczyk operator and the Interval Krawczyk operator through the utilization of numerical examples. Our findings demonstrate the superiority of the proposed technique.

Keywords: Polynomial B-spline, finding roots, Interval Krawczyk, Polynomial systems.

I. INTRODUCTION

Efficient computational techniques are essential in engineering applications, such as analysis of electrical networks and computer-aided design. One particular challenge is the need to locate all roots of nonlinear polynomial equations in *v* variables, which are confined to an **y** dimensional box.

Different pruning operator including Krawczyk operator decrease iterations. These operators require derivatives to evaluate interval enclosures. Interval approaches take time to find polynomial derivatives. For solving nonlinear polynomial equations, a combination of Interval Krawczyk with domain subdivision is presented in [7], [8] for B-spline and Bernstein basis respectively.

A suggested algorithm was based on B-spline expansion strategy mixed with B-spline Krawczyk operator is utilized to determine the solutions of a set of equations involving polynomials i.e. roots.

To implement the B-spline expansion method, the power-form polynomial of the objective function is converted into a B-spline with same degree (m). The coefficients of B-spline expansion then provide a lowest and greatest bound on the range of the objective function.

The present paper is structured in the following manner: In the second section, a concise overview is provided of the B-spline representation of polynomials and the associated domain division technique. In the third section, we provide an explanation of the interval Krawczyk operator and present the algorithm for its B-spline form implementation. In the fourth section, we provide a proposed approach for effectively solving the system of polynomial equations. This algorithm incorporates the B-spline Krawczyk operator as a means of refining and narrowing down the boundaries. In the fifth section, we test the performance B-spline Krawczyk operator (BKO) and the interval Krawczyk operator (IKO) using the two problems, we present our last remarks.

II. BACKGROUND: B-SPLINE EXPANSION

In the first place, we will provide a quick introduction to B-spline expansion. The range of in power from polynomial is obtained by using the B-spline expansion. After that, the B-spline shape is used as the foundation for the primary zero finding procedure in section 3.

So as to acquire the B-spline expansion, we follow the approach described in [7] and [6]. Consider $F(x_1, \cdots x_v)$ represent a multivariate polynomial in v real variables, where the polynomial has the largest degree $(d_1 + \cdots + d_v)$ (1).

$F(x_1, \cdots x_v) = \sum_{p_1=0}^{d_1} \cdots \sum_{p_v=0}^{d_v} c_{p_1 \cdots p_v} x_1^{p_1} \cdots x_v^{p_v}.$ (1)

2.1 Univariate polynomial

Lets consider univariate polynomial case first, (2)

$F(x) = \sum_{p=0}^{d} c_v t x^v, x \in [a, b],$ (2)

For a given degree m, this is equivalent to an order of *m*+1. The B-spline expansion is defined on a compact interval I=[a,b], where the condition $m \geq d$ holds. The splines with a degree of *m* on a partition of the uniform grid is referred to as the Periodic or Closed knot vector, and it is denoted by the letter, $\boldsymbol{w}$, and denoted as $\Omega_m(\mathrm{J}, \boldsymbol{w})$, and $\boldsymbol{w}$ is given as,

$\boldsymbol{w} := \{x_0 < x_1 < \cdots < x_{s-1} < x_s\}.$ (3)

The value of $x_j := a + jz, 0 \leq j \leq s,$ where s denotes number segments of B-spline and $z := (b - a)/s.$

Let's say that $\boldsymbol{N}_q$ represents the space occupied by splines of degree q. The degree q splines with C^{q-1} continue on $[a, b]$ and $\boldsymbol{w}$ as knot vector is thus designated by the following notation:

$\Omega_q(I, \boldsymbol{w}) := \{\Omega \in C^{q-1}(I) : \Omega | [z_j, z_{j+1}] \in \boldsymbol{N}_q, j = 0, \cdots, s-1\}.$ (4)

Since $\Omega_q(I, \boldsymbol{w})$ is $(s + q)$ dimension linear space [8]. To provide a foundation for locally supported splines, $\Omega_q(I, \boldsymbol{w})$, we required some extra knots $z_{-q} \leq \cdots \leq z_{-1} \leq a$ and $b \leq z_{s+1} \leq \cdots \leq z_{s+q}$ clamed at the ends of knot vector which are called as Clamped knot vectors, (5). Elements of Open or Clamped knot vector $\boldsymbol{w}$ is obtained as $z_j := a + ju,$

$\boldsymbol{w} := \{z_{-q} \leq \cdots \leq z_{-1} \leq a = z_0 < z_1 < \cdots < z_{s-1} < b = z_s \leq z_{s+1} \leq \cdots \leq z_{s+q}\}.$ (5)

The B-spline basis $\{B_j^q(z)\}_{j=1}^{s-1}$ of $\Omega_q(I, \boldsymbol{w})$ is defined in terms of divided differences:

$B_j^q(z) := (z_{j+q} - z_j)[z_j, z_{j+1}, \cdots, z_{j+q+1}](. - z)_+^q,$ (6)

where $(.)_+^q$ represent degree truncation. This can be simply shown as

$B_j^q(z) := \Omega_d\left(\frac{z-a}{h} - i\right), -q \leq j \leq s-1,$ (7)

where

$$\Delta_q(z) := \frac{1}{q!}\sum_{i=0}^{q+1}(-1)^i \binom{q+1}{v} (z - v)_+^q, \quad (8)$$

$B_j^q(z) := (z_{j+q} - z_j)[z_j, z_{j+1}, \cdots, z_{j+q+1}](. - z)_+^q,$ is degree q basis function. The expression for basis in B-spline form is facilitated by following Cox-deBoor recursion formula,

$B_j^q(z) := \beta_{j,q}(z) B_j^{q-1}(z) + \left(1 - \beta_{j+1,q}(z)\right) B_{j+1}^{q-1}(z), \ q \geq 1,$ (9)

where

$$\beta_{j,q}(z) = \begin{cases} \frac{z - x_j}{z_{j+q} - z_j}, & \text{if } z_j \leq z_{j+q}, \\ 0, & \text{otherwise}, \end{cases} \quad (10)$$

and

$$B_j^0(z) := \begin{cases} 1, & \text{if } z \in [z_j, z_{j+1}), \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The spline basis set $\{B_j^q(z)\}_{j=1}^{s-1}$ has the following desirable characteristics:

1. Every $B_j^q(z)$ is greater than zero on $[z_j, z_{j+q+1}]$.
2. The spline basis set $\{B_j^q(z)\}_{j=1}^{v-1}$ shows a unity partition, i.e.

$$\sum_{j=1}^{v-1} B_j^q(z) = 1.$$

The following relation may be used to express the $\{z^l\}_{l=0}^m$ in (2) in terms of B-spline.

$$z^l := \sum_{r=-q}^{v-1} \pi_r^{(l)} B_r^q(z), l = 0, \cdots, m, \tag{12}$$

where the symmetric polynomial $\pi_r^{(l)}$ defined as

$$\pi_r^{(l)} := \frac{\text{Sym}_s(r+1, \cdots, r+q)}{s^l \binom{q}{l}}, \; l = 0, \cdots, m. \tag{13}$$

Then by substituting (12) in (2) we get the power form polynomial (3)'s B-spline extension as follows:

$$F(z) := \sum_{p=0}^{m} c_p \sum_{r=-q}^{v-1} \pi_r^{(l)} B_r^q(z) = \sum_{r=-q}^{v-1} \left[\sum_{p=0}^{m} c_p \pi_r^{(l)} \right] B_r^q(z) = \sum_{r=-q}^{s-1} D_n B_r^q(z), \tag{14}$$

where

$$D_n := \sum_{p=0}^{m} c_p \pi_r^{(l)}. \tag{15}$$

2.2 Multivariate polynomial case

Let us now investigate B-spline expansion of following power form multivariate polynomial (17),

$$P(z_1, \cdots, z_v) := \sum_{g_1=0}^{k_1} \cdots \sum_{g_v=0}^{k_v} c_{g_1 \cdots g_v} z_1^{k_1} \cdots z_v^{k_v} = \sum_{g \le k} a_g z^k, \tag{16}$$

where $\boldsymbol{g} := (g_1, \cdots, g_v)$ and $\boldsymbol{k} := (k_1, \cdots, k_v)$. Substituting (12) for each z^k, (16) may also be expressed as

$$F(z_1, z_2, \dots, z_v) = \sum_{l_1=0}^{m_1} \dots \sum_{l_v=0}^{m_v} c_{l_1 \dots l_v} \sum_{u_1=-q_1}^{k_1-1} \pi_{u_1}^{(l_1)} B_{u_1}^{q_1}(z_1) \; \dots \sum_{u_v=-q_v}^{k_v-1} \pi_{u_v}^{(l_v)} B_{u_v}^{q_v}(z_v),$$

$$= \sum_{u_1=-q_1}^{k_1-1} \dots \sum_{u_v=-q_v}^{k_v-1} \left(\sum_{l_1=0}^{m_1} \dots \sum_{l_v=0}^{m_v} c_{l_1 \dots l_v} \pi_{u_1}^{(l_1)} \dots \pi_{u_v}^{(l_v)} \right) B_{u_1}^{q_1}(z_1) \dots B_{u_v}^{q_v}(z_v), \tag{17}$$

$$= \sum_{u_1=-q_1}^{k_1-1} \dots \sum_{u_v=-q_v}^{k_v-1} D_{u_1 \dots u_v} B_{u_1}^{q_1}(z_1) \dots B_{u_v}^{q_v}(z_v),$$

we can write (17) as

$$F(z) := \sum_{u \le k} D_u B_u^k(z). \tag{18}$$

where $u := (u_1, \cdots, u_v)$ and D_u is B-spline coefficient given as

$$D_{u_1 \dots u_v} = \sum_{l_1=0}^{m_1} \dots \sum_{l_v=0}^{m_v} c_{l_1 \dots l_v} \pi_{u_1}^{(l_1)} \dots \pi_{u_v}^{(l_v)}. \tag{19}$$

Equation (18) gives B-spline expansion of equation (17). A polynomial derivative in a specific direction may be determined by using the values of D_u, these are the coefficients of the equation (18) for $\mathbf{y} \subseteq I$. The derivative of $F(x)$ in direction x_r is represented by equation (21).

$$F_r'(\mathbf{y}) = \frac{m_r}{\mathbf{w}_{s+m_r+1} - \mathbf{w}_{s+1}} \times \sum_{I \le m_{r,-1}} [D_{s_{r,1}}(\mathbf{y}) - D_s(\mathbf{y})] B_{m_{r,-1},s}(x),\ 1 \le r \le v, x \in \mathbf{y}, \tag{20}$$

If $\mathbf{w}$ is a knot vector then partial derivative $F_r'(\mathbf{y})$ gives the bound of the range enclosure for the derivative of F with respect to $\mathbf{y}$. In their work, Lin and Rokne proposed (14) for symmetric polynomials, using a closed or periodic knot vector. As a result of the modification in the knot vector from (4) to (6), we suggest a revised formulation of (14) in the subsequent manner,

$$\pi_u^{(l)} := \frac{\mathrm{Sym}_v(u+1, \cdots, u+q)}{\binom{q}{l}}. \tag{21}$$

2.3 B-spline range enclosure property

$$F(z) := \sum_{i=1}^{m} D_i B_i^q(z), z \in \mathbf{y}. \tag{22}$$

Consider the B-spline expansion (23) representing the polynomial $g(t)$. Let $\bar{g}(\mathbf{y})$ indicate range of $g(t)$ on subbox $\mathbf{y}$. The array $\mathrm{D}(\mathbf{y})$ consists B-spline coefficients,

$$\bar{g}(\mathbf{y}) \subseteq \mathrm{D}(\mathbf{y}) = [\min \mathrm{D}(\mathbf{y}), \max \mathrm{D}(\mathbf{y})]. \tag{23}$$

The above equation show that interval of the lowest and maximum values of $\mathrm{D}(\mathbf{y})$ gives bound for the range of equation (17) g on $\mathbf{y}$.

2.4 Domain division procedure

The enclosure of range achieved by B-spline expansion may be enhanced by using the technique of domain division of subbox $\mathbf{y}$. Let

$y := [\underline{y}_1, \bar{y}_1] \times \cdots \times [\underline{y}_r, \bar{y}_r] \times \cdots \times [\underline{y}_v, \bar{y}_v]$,

the box that has to be consider for domain subdivison in the rth direction $(1 \leq r \leq v)$. It results in two subboxes $\mathbf{y}_A$ and $\mathbf{y}_B$ as follows

$\mathbf{y}_A := [\underline{y}_1, \bar{y}_1] \times \cdots \times [\underline{y}_r, m(\mathbf{y}_r)] \times \cdots \times [\underline{y}_v, \bar{y}_v]$,

$\mathbf{y}_B := [\underline{y}_1, \bar{y}_1] \times \cdots \times [m(\mathbf{y}_r), \bar{y}_r] \times \cdots \times [\underline{y}_v, \bar{y}_v]$,

where $m(\mathbf{y}_r)$ is a midpoint of $[\underline{\mathbf{y}}_r, \overline{\mathbf{y}}_r]$.

III. B-SPLINE KRAWCZYK OPERATOR ALGORITHM

The method for the B-spline Krawczyk operator suggested in this study is founded upon the interval Krawczyk used as a pruning operator. The purpose of introducing this approach is to minimize the total iterations required. Interval Krawczyk pruning operator is defined as follows,

$M = b - Nf(b) + J - NI(a)(a - b)$.

Where N is a real matrix that is nonsingular as a precondition, i.e., $N = \left(mid\ \ I(a)\right)^{-1}$ and where I is Jacobian matrix evaluated on the interval a, and where b denotes the interval's midpoint, i.e., $b = mid(a)$. Interval computation Krawczyk operator requires midpoint nonlinear polynomial equation evaluation $f(b)$ and Jacobian matrix is computed across the given interval $\mathbf{y}$. The following is summary of the suggested B-spline Krawczyk operator algorithm,

Step 1: Consider a domain box or interval, $\mathbf{x}$ and a cell structure $D_c(x)$, consisting of B-spline coefficients $D_i(x)$ of given polynomial systems defined on the domain x.

Step 2: Then we compute the interval midpoint $y$as $y = mid(x)$.

Step 3: Next, we evaluate all polynomial function at the midpoint $f_i(y)$.

Step 4: The B-spline derivative technique is utilized to calculate the derivatives of polynomial systems in each component direction, and it is denoted as an interval value, $I(x)$.

Step 5: Compute the determinant of $mid(I(x))$if it is less than ε, then generate two subboxes . Chose the subdivision direction along the longest direction of $x$and the subdivision point as the midpoint. Subdivide $x$into two subboxes x_1 and x_2 such that $x = x_1 \cup x_2$. Compute the B-spline coefficients for both the subboxes and store both these items $\{x_1, D(x_1)\}$and $\{x_2, D(x_2)\}$ into the list $\mathcal{L}$.

Step 6: If the determinant of $mid(I(x))$ is greater than ε compute preconditioner M as $M = \left(mid\left(I(x)\right)\right)^{-1}$.

Step 7: Obtain B-spline Krawczyk operator, M as $M = b - Nf(b) + J - NI(a)(a - b)$ where J is unity matrix of size $(v \times v)$ and v is the number of variables.

IV. ZERO FINDING ALGORITHM

The algorithm based on B-spline Krawczyk operator for finding the roots of nonlinear polynomial equations with *v* variables that fit inside a box with *v* dimensions. The following is summary of algorithm

Step 1: Consider A_c consisting of the array of coefficients a_i of the *n* polynomial.

Step 2: Consider N_c, consisting N_i, which contains the degree of each variable in a polynomial.

Step 3: Then we compute the $D_i(x)$ for the polynomials under consideration on box $\mathbf{x}$.

Step 4: Consider a list $\mathcal{L}$ as $\mathcal{L} \leftarrow \{x, D_i(x)\}$ and $\mathcal{L}^{Sol}$ to empty as solution list.

Step 5: Start iteration, check where $\mathcal{L}$ is empty or not, if empty then implement step 14 else take last element from $\mathcal{L}$, consider this last element is $\{b, D_i(b)\}$ and remove this element from $\mathcal{L}$.

Step 6: Test the feasibility of the box b, for the enclosure of roots. If $any(min(D_i(b))) > 0$ else if $any(max(D_i(b))) < 0$ then discard the box b as it does not enclose the roots and implement step 5 else implement step 7.

Step 7: Take the new box b as root. If $width(b) < \varepsilon$ then store $b$in the list $\mathcal{L}^{Sol}$ and implement step 5 else implement step 8.

Step 8: Compute M, using B-spline Krawczyk operator algorithm.

Step 9: Next compute the updated bound values as $b_{\text{new}} = b \cap M$.

Step 10: Examine the revised bound value's validity. If $b_{\text{new}} = \emptyset$ then delete the item $\{b, D_i(b)\}$ and implement step 5 else implement step 11.

Step 11: Subdivision confirmation, if the variable boundaries are reduced by more than 20% in any variable direction, evaluated as $any(width(b_{\text{new}})) < 0.8 \times (width(b))$ then implement step 12 else implement step 13.

Step 12: Compute $D_i(b_{\text{new}})$ on b_{new} and update the list $\mathcal{L}$ with item $\{b_{\text{new}}, D_i(b_{\text{new}})\}$.

Step 13: Generate two items. The procedure for picking the direction of subdivision along the longest axis may be summed up as follows for b and the middle point of the subdivision. Divide b into b_1 and $b_2$then enter both the items $\{b_1, D(b_1)\}$and $\{b_2, D(b_2)\}$ into the list $\mathcal{L}$and go to step 5.

Step 14: Return all the roots found above.

V. NUMERICAL RESULTS

The calculations are performed on a PC having i3-370M processor, and the algorithms are programmed using MATLAB [10].

We evaluate and compare the performance of the proposed prune operator i.e. (BKO) and the interval Krawczyk operator (IKO) using the two problems. The efficacy metrics consist of the number of iterations and the amount of time required for computation (in seconds).

Example 1:

This is an example from [19], [20], the polynomial system is given by

$$\begin{aligned}
&1 + x_1 + x_2 + x_3 + x_4 = 0,\\
&x_1 + x_1x_2 + x_2x_3 + x_3x_4 + x_4 = 0,\\
&x_1x_2 + x_1x_2x_3 + x_2x_3x_4 + x_3x_4 + x_4x_1 = 0,\\
&x_1x_2x_3 + x_1x_2x_3x_4 + x_2x_3x_4 + x_3x_4x_1 + x_4x_1x_2 = 0,
\end{aligned}$$

and the bounds on the variables are $x_2 = [0.95, 1.05]$, $x_3 = [-2.65, -2.6]$, $x_1 = [0.95, 1.05]$, $x_4 = [-0.4, -0.37]$.

The results are tabulated in Table I.

Table I: Roots value and comparison of performance between BKO and IKO.

Roots			
x_1	x_2	x_3	x_4
1	1	-2.6180	-0.3820

	Number of iterations	Computation Time (Sec.)
BKO	4	1.248
IKO	6	1.336

Example 2:
This example is taken from [2]. The system of polynomial equations is

$$5x_1^9 - 6x_1^5x_2^2 + x_1x_2^4 + 2x_1x_3 = 0$$
$$-2x_1^6x_2 + 2x_1^2x_2^3 + 2x_2x_3 = 0$$
$$x_1^2 + x_2^2 - 0.265625 = 0$$

and the bounds on the variables are $x_1 = [0.45, 0.5]$, $x_2 = [0.2, 0.24]$, $x_3 = [0, 0.03]$.

The results are tabulated in Table II.

Table II: Roots value and comparison of performance between BKO and IKO.

Roots		
x_1	x_2	x_3
0.4670	0.2180	0
	Number of iterations	**Computation Time (Sec.)**
BKO	4	0.728
IKO	5	1.1648

VI. CONCLUSION

This work introduces a unique approach for determining all solutions of a system of nonlinear polynomial equations in s variables that are contained within an *v*-dimensional box. Two examples were shown to demonstrate the superior performance of the B-spline Krawczyk operator. The presented approach demonstrates its ability to accurately surround the roots of polynomial systems in a limited amount of iterations and computational time, as compared to the interval Krawczyk operator.

REFERENCES

[1] Floudas CA, Pardalos PM. *Handbook of Test Problems in Local and Global Optimization.* Kluwer Academic Publishers, Dordrecht, The Netherlands; 1999.

[2] Horst R, Pardalos PM. *Handbook of Global Optimization*. Kluwer Academic, Netherlands; 1995.

[3] Hansen E, Walster G. *Global Optimization Using Interval Analysis: Revised and Expanded*. Vol 264. (2, ed.). Marcel Dekker, New York; 2004.

[4] Jaulin L. Applied Interval Analysis: with Examples in Parameter and State Estimation. *Robust Control Robot Springer*. 2001

[5] Stahl V. *Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations. PhD Thesis, Johannes Kepler University, Linz.*; 1995.

[6] Lin Q, Rokne JG. Interval approximation of higher order to the ranges of functions. *Comput Math with Appl*. 1996;31(7):101-109.

[7] Lin Q, Rokne JG. Methods for bounding the range of a polynomial. *J Comput Appl Math*. 1995;58:193-199.

[8] DeVore RA, Lorentz GG. *Constructive Approximation*. Vol 303. Springer Science & Business Media, Berlin; 1993.

[9] Patil B. V., Nataraj P. S. V., Bhartiya S. Global optimization of mixed-integer nonlinear (polynomial) programming problems: the Bernstein polynomial approach. *Computing*. 2012;94(2-4):325-343.

[10] Mathworks Inc. *MATLAB Version 8.0.0.783 (R 2012 B)*. Inc., Natick, Massachusetts, United States; 2012.